National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL-339 (r.86/06)

Canada

THE UNIVERSITY OF ALBERTA

DYNAMIC SIMULATION OF THE CONTROL BEHAVIOR OF MINERAL GRINDING CIRCUITS

by

GREGORY BURKE McDOUGALL

©

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE
IN
PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA
SPRING 1987

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-37780-1

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR	GREGORY BURKE McDOUGALL
TITLE OF THESIS	DYNAMIC SIMULATION OF THE CONTROL BEHAVIOR OF MINERAL GRINDING CIRCUITS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1987

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) G.B. McDougall

PERMANENT ADDRESS:

3148 Linden Dr. S.W.
Calgary, Alberta
Canada T3E 6C4

DATED January 2 1987

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled DYNAMIC SIMULATION OF THE CONTROL BEHAVIOR OF MINERAL GRINDING CIRCUITS submitted by GREGORY BURKE McDOUGALL in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in PROCESS CONTROL.

....
Supervisor

....

...Brian C. Flintoff....

Date...12 MARCH 1987..

To the Future

Abstract

Over the past three decades, the mineral processing industry has shown an increasing interest in the area of process control. Although computer simulation has been used from the mid 1960's to aid mineral engineers in the design and steady state analysis of mineral processes, dynamic simulation for studying process control applications only began to gain popularity from the mid 1970's.

This work is concerned with the simulation of the dynamic and control behavior of mineral grinding circuits. The primary purpose is to demonstrate the usefulness of a dynamic grinding circuit simulator to the mineral processing engineer for tasks such as control strategy evaluation. A general dynamic simulation package, MINSIM, is developed based on the DYFLO2 programming approach of Franks (1982) and earlier work at the University of Alberta. The modelling theory used in the simulator design is described in detail along with several advanced control schemes such as multivariable time delay compensation and noninteracting control. The MINSIM software package is used to demonstrate the dynamic behavior of the Lake Dufault grinding circuit and analyze the performance of both open and closed circuit grinding operations utilizing various continuous and discrete control strategies.

It was found that although time delay compensation had little effect on control performance due to model mismatch,

the noninteracting control strategy did prove to be of value.

Acknowledgment

I would like to express my deep gratitude and thanks to many people who either directly or indirectly influnced the course of this work. Foremost is Dr. R.K. Wood, for his patience, guidance, understanding, and encouragement throughout the past two years. His supervision, and Dr. B.C. Flintoff's keen interest in process control applications in the mineral industry, are directly responsible for the conception of this project. Fellow graduate students deserving special mention include R. Broenink, J. Langman, D. Shook, A. Vien, and P. Vermeer for the endless hours spent fielding my sometimes stupid questions and providing ears when needed. Of course, this list is not complete without thanking the entire graduate student class and staff of the Chemical Engineering Department.

The financial assistance by both the Department of Chemical Engineering at the University of Alberta and the Natural Sciences and Engineering Research Council of Canada is also acknowledged and greatly appreciated.

Finally, special recognition is given to my fiance, Kimberly, without whom I would not have survived this ordeal.

Table of Contents

Chapter Page

Abstract v

Acknowledgment vii

1. Introduction 1

1.1 Introduction 1

1.2 Literature Survey 2

1.2.1 Process Control Applied to Mineral Processing Circuits 2

1.2.2 Modelling Mineral Processes 5

1.2.3 Dynamic Simulation Software for Mineral Processes 6

1.2.4 Advanced Control System Simulation Studies with Applications in the Mineral Processing Industry 9

1.2.5 Grinding Circuit Control and Simulation at the University of Alberta 9

1.3 Thesis Organization 10

2. Control Theory 12

2.1 Introduction 12

2.2 Proportional, Proportional Integral, and Proportional Integral Derivative Feedback Controllers 13

2.2.1 Feedback Control 13

2.2.2 Proportional Control 15

2.2.3 Proportional Integral Control 17

2.2.4 Proportional Integral Derivative Control .22

2.3 Discrete PID Controllers 25

2.3.1 Discrete PID Algorithms 25

2.3.2 Discrete PID Structures 28

2.4 Feedforward Controllers 31

2.5 Lead-Lag Compensators 36
2.6 Control of a Multivariable System 39
2.6.1 Multiloop Control 41
2.6.2 Interaction Analysis 47
2.6.3 Decoupling Controllers 53
2.6.4 Time Delay Compensation 69
2.7 Adaptive Control 77
2.7.1 Multivariable Self-Tuning Control Law 77
2.7.2 Parameter Estimation 79
3. Grinding Circuit Dynamic Model 80
3.1 Introduction 80
3.2 Grinding Mills 81
3.2.1 Grinding Machinery 81
3.2.2 Comminution 84
3.2.3 Grinding Mill Model 88
3.3 Hydrocyclone Classifiers 95
3.3.1 Classification 95
3.3.2 Product Size 98
3.3.3 Hydrocyclone Models 100
3.4 Pump and Sump Model 104
3.4.1 Pump Design 104
3.4.2 Pump Model 107
3.4.3 Sump Model 109
3.5 Conveyor and Pipelines 111
3.5.1 Fixed Time Delays 111
3.5.2 Variable Time Delays 113
3.6 Load Disturbance Modelling 116

3.6.1 Grinding Circuit Disturbance Inputs 116

3.6.2 Simulation of Ore Hardness Changes 117

4. Grinding Circuit Simulator Description 122

4.1 Introduction 122

4.2 Simulation Language Description 124

4.2.1 The DYFLO2 Subroutine Library 124

4.2.2 The Mineral Grinding Circuit Unit Operation Subroutine Library 125

4.2.3 Self-Tuning Controller Subroutine Library 129

4.3 Simulator Structure 129

4.3.1 The Closed Circuit Grinding Simulator ...136

4.3.2 Open Circuit Grinding Simulator 146

4.4 User Interface Description 155

4.4.1 The Closed Circuit Grinding Simulator ...156

4.4.2 The Open Circuit Grinding Simulator 164

4.5 Update Procedure 169

4.5.1 Subroutine Library Update 171

4.5.2 Simulator Executive Update 172

5. Lake Dufault Grinding Circuit Simulation Study 176

5.1 Introduction 176

5.2 Grinding Circuit Description 177

5.3 Open Loop Response 188

5.4 Linear Model Representation 196

5.4.1 The Linear Model 201

5.4.2 Controlled-Manipulated Variable Transfer Function Changes 212

5.4.3 Ore Hardness Disturbance Transfer Functions Changes 216

5.5 Estimation of PI Controller Constants 219
5.6 Multiloop Control 225
5.6.1 Type I Control Behavior 225
5.6.2 Type II Control Behavior 241
5.7 Delay Compensator Design 259
5.8 Time Delay Compensated Behavior 263
5.9 Multivariable Control 284
5.9.1 Interaction Analysis 292
5.9.2 Decoupler Design 293
5.10 Noninteracting Control Behavior 297
5.11 Adaptive Controller Design 312
5.12 Adaptive Control Behaviour 316
6. Digital Control Simulation Study 325
6.1 Introduction 325
6.2 Open Circuit Grinding Operation Description ...325
6.3 Linear Model 327
6.4 Continuous Control System Analysis 330
6.5 Discrete Control System Analysis 336
7. Conclusions and Recommendations 351
7.1 Conclusions 351
7.2 Recommendations 357
8. Nomenclature 361
9. REFERENCES 365
Appendix A: Subroutine Library Descriptions 377
Appendix B: DYFLO2.LIB Subroutine Verification 403
Appendix C: A Review of some Numerical Techniques with Consideration to DYFLO/DYFLO2 429
Appendix D: New DYFLO2.LIB Subroutine Details 446

Appendix E: Lead/Lag Decoupling Analysis 479

Appendix F: Time Delay Compensation Analysis 502

Appendix G: Sample Terminal Sessions 527

List of Tables

Table Page

4.1 DYFLO2.LIB New Subroutines and Functions............126

4.2 MIN.LIB Subroutines and Functions..................130

4.3 STC.LIB Subroutines and Functions..................131

4.4 DYFLO Executive Program Structure and Function......133

4.5 The Closed Circuit Simulator Subroutines and Functions for the First Level Executive Program.....142

4.6 The Closed Circuit Simulator Subroutines and Functions for the Second Level Executive Programs Specific to Lake Dufault..........................151

4.7 The Open Circuit Grinding Simulator Subroutines and Functions..156

4.8 Self-Tuning Control Option Interactive Input Summary163

4.9 Closed Circuit Simulator Temporary Output File Summary..165

4.10 Closed Circuit Grinding Simulator Menu and Option Summary..167

4.11 Open Circuit Grinding Simulator Menu and Option Summary..170

4.12 Logical Units Referenced by the Closed Circuit Simulator..174

5.1 First Column of the Breakage Matrix for the Lake Dufault Grinding Circuit..........................183

5.2 First Order Selection Constants for the Lake Dufault Grinding Circuit..........................183

5.3 Steady State Particle Size Frequency................185

5.4 Steady State Particle Size Frequency Distribution of the Lake Dufault Hydrocyclone Feed and Product Streams 187

5.5 A Two by Two Transfer Function Model of the Lake Dufault Mineral Grinding Circuit....................223

5.6 Summary of the Simulations for the Lake Dufault Grinding Circuit using the Nonlinear Model and the Type I Variable Pairing Scheme.......................240

5.7 Summary of the Simulations for the Lake Dufault Grinding Circuit using the Linear Model and the Type I Variable Pairing Scheme.......................244

5.8 Summary of the Simulations for the Lake Dufault Grinding Circuit using the Nonlinear Model and the Type II Variable Pairing Scheme......................258

5.9 Summary of the Simulations for the Lake Dufault Grinding Circuit using the Linear Model and the Type II Variable Pairing Scheme......................262

5.11 Summary of the Simulations for the Lake Dufault Grinding Circuit using Time Delay Compensation......277

5.10 Summary of the Simulations for the Lake Dufault Grinding Circuit using Time Delay Compensation and the Type I Variable Pairing Scheme..................277

5.12 Summary of the Simulations for the Lake Dufault Grinding Circuit using Time Delay Compensation and the Type II Variable Pairing Scheme.................290

5.13 First Order Transfer Function Model Representation of the Lake Dufault Grinding Circuit Including the Decoupling Controllers..............................298

5.14 Summary of the Simulations for the Lake Dufault Grinding Circuit using Decoupling Control with the Type I Variable Pairing Scheme.......................306

5.15 Summary of the Simulations for the Lake Dufault Grinding Circuit using Decoupling Control with the Type II Variable Pairing Scheme....................313

5.16 Self-Tuning Controller Polynomial Coefficient Numbers for Type I and Type II Configurations.......316

5.17 Self-Tuning Controller Q Weighting Polynomial Parameters from the well tuned Continuous PI Controller Constants................................317

6.1 Open Circuit Grinding Simulation Equilibrium........331

6.2 Ziegler Nichols Quarter Decay Ratio Tuning Formulae.335

6.3 Ziegler Nichols Controller Constants................336

6.4 Modified Ziegler Nichols Digital Controller Tuning Formulae after Adel et al. (1983)....................344

6.5 Modified Ziegler Nichols Digital Controller Constants...345

B.1 Parameter Description for the DYFLO2 Routine TDL....417

B.2 Internal Variable Description for the DYFLO2........418

B.3 Summary of TDL (DYFLO2) Logic.........................420

B.4 Sample Output for the Fixed Time Delay Simulation using First Order Integration.......................426

B.5 Sample Output for the Fixed Time Delay Simulation using Second Order Integration......................427

B.6 Sample Output for the Fixed Time Delay Simulation using Fourth Order Integration......................428

List of Figures

Figure Page

2.1 Block Diagram for a General SISO Feedback Control Loop Configuration....................................14

2.2 Typical Closed Loop System Response to a Unit Step Change in Setpoint Showing the Effect of Increased Proportional Gain on Stability and Offset............17

2.3 Response of a PI Controller to a Unit Step in the Input Error Signal.................................18

2.4 Block Diagram for an Ideal PI Controller.............20

2.5 Block Diagram Showing Reset Feedback Implementation of a PI Controller with an Output Limiter for Reset Windup Protection...................................22

2.6 Block Diagram for a PID Controller with Derivative Action Only on the Output Signal Using a Lead Lag Approximation of the Ideal Derivative Action.........24

2.7 Block Diagram for a Classical Discrete PID Controller29

2.8 Block Diagram for a Discrete PID Controller with Derivative Action Only on the Output Signal..........29

2.9 Block Diagram for a Discrete PID Controller with Integral Action Only on the Setpoint Signal..........31

2.10 Block Diagram for a Feedforward - Feedback Control Scheme...33

2.11 A General Multivariable Process.....................40

2.12 A General Two by Two Interacting Process.............42

2.13 A Classical Multiloop Control System Ignoring Process Interactions.................................42

2.14 Equivalent Representation of a Block Diagram for a Multiloop Control System........................43

2.15 Block Diagram for a General Noninteracting Control Scheme..56

2.16 Equivalent Representation of a Block Diagram for a General Noninteracting Control Scheme................56

2.17 Block Diagram for the Boksenbom and Hood Noninteracting Control Scheme.........................62

2.18 A Block Diagram for the Zalkind Noninteracting Control Scheme.......................................66

2.19 Block Diagram for a SISO Control System with Process and Measurement Time Delays..................71

2.20 Block Diagraom for a SISO Control System with a Smith Predictor Time Delay Compensator...............71

2.21 Block Diagram for a General Multivariable Multidelay Compensator Feedback Control System.......74

2.22 Block Diagram for a Two by Two Time Delay Compensated Control System...........................76

3.1 Schematic Diagram of a Typical Rod Mill..............82

3.2 Schematic Diagram of a Typical Ball Mill.............82

3.3 Sieve Tray Analysis for Discrete Particle Size Distribution Reference Points........................86

3.4 Schematic Diagram of a Typical Hydrocyclone..........96

3.5 Typical Relationship Between Particle Size and Hydrocyclone Underflow Recovery.....................98

3.6 Typical Reduced Efficiency Curve for a Hydrocyclone.100

3.7 The Effect of Variations in γ on the Shape of the Reduced Efficiency Curve....................103

3.8 Flat Pump Curve....................107

3.9 Steep Pump Curve....................107

3.10 Computer Representation of a Bucket Brigade.........112

3.11 Circular Buffer Schematic....................112

3.12 Variable Time Delay Schematic Diagram...............114

4.1 A Data Flow Diagram for a General DYFLO2 Executive Program....................134

4.2 A Program Flow Chart for a General DYFLO2 Executive Program....................135

4.3 First Level Data Flow Diagram for the Closed Circuit Grinding Simulator....................138

4.4 Second Level Data Flow Diagram for the Closed Circuit Grinding Simulator....................139

4.5 Program Structure Diagram for the Main Executive of the Closed Circuit Grinding Simulator...............141

4.6 Program Structure Diagram for the Open Loop Executive of the Closed Circuit Grinding Simulator..145

4.7 Program Structure Diagram for the Multiloop Type I and Type II Control Scheme Executive for the Open Circuit Grinding Simulator....................147

4.8 Program Structure Diagram for the Type I and Type II Noninteracting Control Scheme Executive for the Open Circuit Grinding Simulator....................148

4.9 Program Structure Diagram for the Type I and Type II Time Delay Compensation Control Scheme Executive for the Open Circuit Grinding Simulator.............149

4.10 Program Structure Diagram for the Type I and Type II Self Tuning Control Scheme Executive for the Open Circuit Grinding Simulator.....................150

4.11 Data Flow Diagram for the Open Circuit Grinding Simulator......................................154

4.12 Program Structure Diagram for the Executive of the Open Circuit Grinding Simulator.....................156

5.1 Schematic Diagram of the Lake Dufault Closed Circuit Grinding Operation.........................178

5.2 Simulation Flowsheet for the Lake Dufault Grinding Circuit...181

5.3 Open Loop Response of COS to a ±10% Step in FOF.....190

5.4 Open Loop Response of COS to a ±10% Step in SWF.....191

5.5 Open Loop Response of COS to a ±50% Step in HRD.....192

5.6 Open Loop Response of SPL to a ±5% Step in FOF......193

5.7 Open Loop Response of SPL to a ±5% Step in SWF......194

5.8 Open Loop Response of SPL to a ±50% Step in HRD.....195

5.9 A Block Diagram of a 2x2 Interacting System for the Lake Dufault Grinding Circuit.........................198

5.10 A Typical Process Reaction Curve Showing the First Order Plus Time Delay Transfer Function Characterization Points...............................200

5.11 Characterization of COS for a Negative FOF Step Disturbance.......................................202

5.12 Characterization of COS for a Positive SWF Step Disturbance....................................202

5.13 Characterization of COS for a Positive HRD Step Disturbance....................................203

5.14 Characterization of SPL for a Positive FOF Step Disturbance....................................203

5.15 Characterization of SPL for a Positive SWF Step Disturbance....................................204

5.16 Characterization of SPL for a Positive HRD Step Disturbance....................................204

5.17 Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±10% Step in FOF........206

5.18 Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±10% Step in SWF........207

5.19 Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±50% Step in HRD........208

5.20 Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±10% Step in FOF........209

5.21 Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±10% Step in SWF........210

5.22 Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±50% Step in HRD........211

5.23 Open Loop Response of COS using the Modified Linear Model to ±10% Step Changes in FOF..................214

5.24 Open Loop Response of COS using the Modified Linear Model to ±10% Step Changes in SWF..................215

5.25 Characterization of COS for a Negative HRD Step Disturbance....................................217

5.26 Characterization of SPL for a Negative HRD Step Disturbance..217

5.27 Open Loop Response of COS using the Average Linear Circuit Models to ±50% Step in HRD..................220

5.28 Open Loop Response of SPL using the Average Linear Circuit Models to ±50% Step in HRD..................221

5.29 A Block Diagram of the Lake Dufault Grinding Circuit Transfer Function Model......................222

5.30 Block Diagram Showing the Type I Control System Configuration......................................226

5.31 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Cohen and Coon PI Controller Settings.........227

5.32 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Controller Settings......................229

5.33 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Controller Settings......................230

5.34 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using the Regulatory Controller Settings...232

5.35 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using the Servo Controller Settings........233

5.36 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using the Servo Controller Settings........235

5.37 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using the Servo Controller Settings.................236

5.38 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using the Servo Controller Settings.................237

5.39 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using the Servo Controller Settings.......................240

5.40 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using the Servo Controller Settings.......................241

5.41 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using the Servo Controller Settings........244

5.42 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using the Servo Controller Settings........245

5.43 Block Diagram Showing the Type II Control System Configuration......................................247

5.44 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Cohen and Coon Controller Settings............246

5.45 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Regulatory Controller Settings..........248

5.46 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Regulatory Controller Settings..........249

5.47 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Tuned Regulatory Controller Settings.250

5.48 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Tuned Servo Controller Settings......251

5.49 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Tuned Servo Controller Settings......252

5.50 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Servo Controller Settings...............253

5.51 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Servo Controller Settings...............254

5.52 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Tuned Servo Controller Settings.....................258

5.53 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Tuned Servo Controller Settings.....................259

5.54 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Tuned Servo Controller Settings......262

5.55 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Tuned Servo Controller Settings......263

5.56 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Time Delay Compensation.........................264

5.57 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Time Delay Compensation.........................265

5.58 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation..............................267

5.59 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Time Delay Compensation..............................268

5.60 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation..............269

5.61 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation..............270

5.62 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation..............271

5.63 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation..............272

5.64 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation with Increased Loop 1 Controller Gain.......................................274

5.65 Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation with Increased Loop 1 Controller Gain....................275

5.66 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Time Delay Compensation.......................279

5.67 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation.............................280

5.68 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation with Increased Loop 2 Gain..281

5.69 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Time Delay Compensation.......................282

5.70 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Time Delay Compensation.............................283

5.71 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation..............285

5.72 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation..............286

5.73 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation..............287

5.74 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation..............288

5.75 Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation with Increased Loop 2 Controller Gain....................289

5.76 Type I Control with Static Decoupling Controllers...295

5.77 Type II Control with Static Decoupling Controllers..295

5.78 Type I Control with Dynamic Decoupling Controllers..296

5.79 Type II Control with Dynmaic Decoupling Controllers.297

5.80 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Static Decoupling Controllers.................300

5.81 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Dynamic Decoupling Controllers................301

5.82 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Static Decoupling Controllers........303

5.83 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling Controllers.......304

5.84 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling with Decreased Loop 2 Gain.......................................305

5.85 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Static Decoupling Controllers.................308

5.86 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Static Decoupling Controllers........309

5.87 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Dynamic Decoupling Controllers................310

5.88 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling Controllers.......311

5.89 Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model using Self-Tuning Control...318

5.90 Type I Lake Dufault Nonlinear Circuit Model Parameter Estimates.................................319

5.91 Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model using Self-Tuning Control...322

5.92 Type II Lake Dufault Nonlinear Circuit Model Parameter Estimates.................................323

6.1 Open Circuit Grinding Mill Schematic Diagram........326

6.2 Closed Loop Block Diagram for the Open Circuit Grinding Mill Simulation Study........................328

6.3 Closed Loop Block Diagram for the Open Circuit Grinding Mill Simuilation Study with Digital Control329

6.4 Open Loop Response of the Open Circuit Grinding Operation to a Step Disturbance....................331

6.5 Closed Loop Response of the Open Circuit Grinding Operation using Proportional Control to a Step Disturbance and Various Analyzer Time Delays........333

6.6 Closed Loop Response of the Open Circuit Grinding Operation using Proportion Plus Integral Control to a Step Disturbance and Various Analyzer Time Delays.333

6.7 Closed Loop Response of the Open Circuit Grinding Operation using P, PI, and PID Control to a Step Disturbance and a 1 Hours Analyzer Time Delay.......337

6.8 Closed Loop Response of the Open Circuit Grinding Operation using a PID Controller with a Rate Amplitude of 0.1....................................337

6.9 Closed Loop Response of the Open Circuit Grinding Operation using Discrete PID Control with Various Sample Intervals....................................340

6.10 Closed Loop Response of the Open Circuit Grinding Operation using Discrete PID Control with Incorrect Sampler Operation....................................339

6.11 Schematic Diagram Showing the Disturbance Input Relative to Sampler Operation......................340

6.12 Pulse Disturbance Input Signal.....................341

6.13 Open Loop Response of the Open Circuit Grinding Operation to a Pulse Disturbance....................342

6.14 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete Proportion Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance.......................................345

6.15 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance...345

6.16 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with Tuned Controller Constants for a Pulse Disturbance........346

6.17 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PID Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance...346

6.18 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete P Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance....348

6.19 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance....349

6.20 Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PID Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance....349

B.1 Subroutine INTCPD Test Executive Program Listing....407

B.2 Analytical vs Numerical Solution Using the Iterative Implicit Integration Technique for an Integration Time Step of 0.1 Units..................406

B.3 Analytical vs Numerical Solution Using the Iterative Implicit Integration Technique for an Integration Time Step of 0.05 Units.................408

B.4 Subroutine GJR Test Executive Program Listing.......412

B.5 Sample Output from Subroutine GJR Test Program......412

B.6 Subroutine ZOH Test Executive Program Listing.......414

B.7 Sample Output from Subroutine ZOH Test Program......415

B.8 DYFLO2 Fixed Time Delay Subroutine TDL Listing......417

B.9 Block Diagram Representation of a Simple Process....425

B.10 Block Diagram Decomposed for DYFLO2 Simulation......424

B.11 Subroutine TDL Test Executive Program Listing.......425

C.1 Mechanics of the Explicit Euler Solution Technique..431

C.2 Graphical Interpretation of the Recursive Form of the Explicit Euler ODE Solution Technique...........433

C.3 Graphical Interpretation of the Recursive Form of the Implicit Euler ODE Solution Technique...........435

D.1 Subroutine CON2X2 FORTRAN IV Source Code Listing....448

D.2 Subroutine CON2X2 Program Flow Chart................449

D.3 Subroutine DISPID FORTRAN IV Source Code Listing....453

D.4 Subroutine DISPID Program Flow Chart................454

D.5 Subroutine DPID2 FORTRAN IV Source Code Listing.....453

D.6 Subroutine I2BY2 FORTRAN IV Source Code Listing.....455

D.7 Subroutine I2BY2 Program Flow Chart.................456

D.8 Subroutine LLCOMP FORTRAN IV Source Code Listing....455

D.9 Subroutine LLCOMP Program Flow Chart................456

D.10 Subroutine LLCDER FORTRAN IV Source Code Listing....460

D.11 Subroutine PTRAIN FORTRAN IV Source Code Listing....460

D.12 Subroutine PTRAIN Program Flow Chart................464

D.13 Subroutine RATCON FORTRAN IV Source Code Listing....463

D.14 Subroutine RATCON Program Flow Chart................465

D.15 Subroutine TDC FORTRAN IV Source Code Listing.......466

D.16 Subroutine TDC Program Flow Chart...................467

D.17 Subroutine ZOH FORTRAN IV Source Code Listing.......466

D.18 Subroutine ZOH Program Flow Chart...................471

D.19 Subroutine INTCPD FORTRAN IV Source Code Listing....470

D.20 Subroutine INTCPD Program Flow Chart................471

D.21 Subroutine NUMJAC FORTRAN IV Source Code Listing....470

D.22 Subroutine NUMJAC Program Flow Chart................476

D.23 Subroutine JFUNC FORTRAN IV Source Code Listing.....475

D.24 Subroutine JFUNC Program Flow Chart.................476

D.25 Subroutine RELAX FORTRAN IV Source Code Listing.....476

D.26 Subroutine RELAX Program Flow Chart.................478

E.1 Lead Time Dominant Lead-Lag Compensator Response to a Step Input.......................................483

E.2 Lag Time Dominant Lead-Lag Compensator Response to a Step Input.......................................483

E.3 Lag Time Dominant Lead-Lag Compensator with a Two Unit Delay...483

E.4 Lead Lag Decoupling DYFLO2 Program Source Code Listing....................................484

E.5 Multiloop PI Control of the Original Process........489

E.6 Multiloop PI Control of the Original Process with Static Decoupling..................................489

E.7 Multiloop PI Control of the Undelayed Process with Dynamic Decoupling.................................492

E.8 Multiloop PI Control of the Undelayed Process.......493

E.9 Multiloop PI Control of the Original Process with Dynamic Decoupling.................................492

E.10 Multiloop PI Control of the Original Process with Dynamic Decoupling, Load Disturbance Response.......496

E.11 Multiloop PI Control of a Process Requiring Predictive Dynamic Decoupling......................496

E.12 Multiloop PI Control of a Process with Slow Loop 2 Direct Transmission Dynamics.........................500

E.12 Multiloop PI Control of a Process with Slow Loop 2 Direct Transmission Dynamics Using Dynamic Decoupling...500

F.1 Block Diagram for Time Delay CompensationSubroutine Verification...503

F.2 Loop 1 Open Loop Response of the Time Delay Compensator to a Step Input..........................505

F.3 Loop 2 Open Loop Response of the Time Delay Compensator to a Step Input..........................505

F.4 Time Delay Compensation DYFLO2 Program Source Code Listing..505

F.5 Demonstration of the Decrease in ISE Sensitivity when Process Time Delays are Present................509

F.6 Multiloop PI Control of the Original Process........511

F.7 Multiloop PI Control of the Time Delay Conpensated Process..513

F.8 Multiloop PI Control of a Process with a Short Off-diagonal Time Delay Element.....................513

F.9 Multiloop PI Control of a Process with a Short Off-diagonal Time Delay Element using Time Delay Compensation.......................................517

F.10 Block Diagram for Decoupling and Time Delay Compensation Simulation..............................516

F.11 Multiloop PI Control of the Original Process with Decoupling and Time Delay Compensation for Loop 1...519

F.12 Multiloop PI Control of the Original Process with Decoupling and Time Delay Compensation for Loop 2...520

F.13 Multivariable Block Diagram Representation for Decoupling and Time Delay Compensation Transfer Function Analysis......................................522

F.14 Multiloop PI Control of a Process with a Short Off-diagonal Element using Decoupling and Time Delay Compensation.................................524

F.15 Block Diagram for Alternate Decoupling with Time Delay Compensation Control System..................526

1. Introduction

1.1 Introduction

Over the past three decades, the mineral processing industry has shown an increasing interest in the area of process control. This is due, in part, to an increasing awareness of the economic advantages a tightly controlled plant offers, including a reduction in energy consumption, the maintenance of a consistent product quality, and increased plant throughput (Herbst and Bascur, 1984). Because the unit operation approach to process control used in the chemical industry is well developed and can be directly applied to mineral processes, mineral engineers are looking to the chemical industry to provide expertise in process control.

The use of dynamic simulation for studying the dynamic and control behavior of mineral processing circuits has only become popular since the mid 1970's. Most of the effort in this area has gone into the development of simulators capable of simulating particular grinding mills and flotation circuts (Flintoff et al., 1985). Only recently have general purpose dynamic simulators that contain models of other unit operations such as crushing and particle size separation along with more complex integrated circuits including auxillary processing equipment such as sumps and pumps become available (Adel, 1982; Flintoff et al. 1985; Rajamani and Herbst, 1980; Ferrara et al., 1984).

The purpose of this thesis is twofold. The primary objective is to demonstrate the usefulness of a dynamic grinding circuit simulator to the mineral processing engineer for purposes such as operator training and control system evaluation. This is achieved by developing the general grinding circuit simulation package, MINSIM, which is based on the DYFLO2 (Franks, 1982) framework and standard mineral process unit operation models (Flintoff et al., 1985; Wong, 1984; Wood and Flintoff, 1982). The underlying objective is to provide the engineer with an insight into some advanced control techniques such as multivariable time delay compensation, noninteracting control, and self tuning control by utilizing the simulator to study the performance of the Lake Dufault closed circuit grinding operation using such schemes. A second, open circuit grinding operation using discrete control is simulated as well.

The next section of this chapter presents a literature survey and a discussion of the previous work done at the University of Alberta. This chapter concludes with an outline of the organization of the thesis.

1.2 Literature Survey

1.2.1 Process Control Applied to Mineral Processing Circuits

Much work has recently been done in the area of process control applications in the mineral industry. Herbst and Bascur (1984) provide a review of the control strategies in use in the mineral industry and give an overview of the more recent trends. Lynch (1984) describes the widely varying control applications and problems currently associated with four different mineral processing areas and discusses future expectations for each.

Specific applications, covering a wide range of circuit configurations and different control strategies, have been presented by many authors. Smith and Lewis (1969) discuss an operator emulator application of a process computer for performing duties such as material balance calculations, and flotation reagent flow rate calculations for control of a flotation circuit at Lake Dufault Mines Limited. Fewings (1976) has reported on the use of a simulation package to design a computer control system for the grinding circuit of a lead-zinc concentrator at Mount Isa Mines Limited. Application of this control system to the plant resulted in a 5% increase in throughput, as predicted from the simulation results. More recently, Lean and Baker (1984) have discussed the computer control system installed at New Broken Hill Consolidated Limited while Allee et al. (1985) have described the specification and installation of a computerized distributed control system for the concentration circuit at Chino Mines Company. The various stages involved in comissioning a microprocessor based multivariable control system for a gold recovery plant have been presented by Gray (1985).

Applications involving sophisticated multivariable control strategies on a pilot scale grinding ciruit have been described by Hulbert et al. (1980), Hulbert and Woodburn (1983) and Barker and Hulbert (1983). In all cases, these researchers used the inverse Nyquist array (INA) multivariable frequency domain technique to design a controller. Simulations conducted by Barker and Hulbert (1983) showed that a reasonable amount of decoupling was achieved which improved the control performance over that obtained using conventional single variable controllers. Lee and Newell (1985) have used a dynamic matrix control (DMC) technique on the circuit described by Hulbert et. al. (1980). They found that although the INA technique produced better control response, the DMC technique proved to be easier to use and provided better control than single loop controllers. Hulbert and Brae (1981) and Hulbert (1983) have applied the INA design technique for the control system design of the full scale milling circuit at the East Driefontein Gold Mine and report on the observed plant behavior. As well, Jamsa et al. (1983) have used the INA technique to develop a multivariable controller for the full scale Vuonos grinding operation. Wyatt-Mair et al. (1980) present a real time control strategy that utilizes a Kalman filter to estimate a state space model of a grinding circuit. This model was used to reliably predict both the measured and unmeasured output variables. Hammoude and Smith (1980) and Su and Yan (1984) have experimented with self tuning control applications on flotation circuits in attempts to overcome some of the control problems associated with these operations.

1.2.2 Modelling Mineral Processes

The mathematical models used in describing the dynamic behavior of mineral processes have been slowly evolving for the past three decades as Wood (1975) documented in a survey of the literature pertaining to the modelling of crushing and grinding circuits published up to 1975. Researchers such as Herbst and Fuerstenau (1980) have created software based on many of these models to aid in the design and sizing of full scale grinding mills from laboratory data. Austin and Klimpel (1984) describe a procedure for estimating several key design parameters from laboratory batch milling data as well. Computer aided methods for estimating the residence time distribution and mass transport characteristics have been discussed by Marchand et al. (1980) and Weller (1980).

Dynamic models for grinding circuits have been developed using both state space and phenomenological (or population balance) modelling approaches. Hinde et al. (1976) and Ragot et al. (1976) present state space models for two different closed circuit grinding configurations. The phenomenological approach has been taken by Lynch (1977), Smith and Guerin (1980), Rajamani and Herbst (1980), and Flintoff et al. (1985). These models are, in general, extensions of phenominological steady state models used in the plant design stage (Flintoff et al., 1985). Methods for identifying the parameters used in the phenomenological models have been described by Fournier and Smith (1972). These researchers also verified that full scale rod mills behave much like laboratory scale ball mills so the same form of model can be used for both types of mill. A third approach to modelling grinding circuits utilizes a time series analysis of plant operating data as described by Romberg and Jacobs (1980). This method has not gained wide acceptance, due to the fact that the models used are empirical in nature and the computer software required to perform the analysis is complex.

1.2.3 Dynamic Simulation Software for Mineral Processes

Computer simulation of mineral processes began in the mid 1960's, with the advent of reasonably powerful digital computers. Most of the early simulators were capable of performing only steady state material balances on specific pieces of equipment in the concentration circuit. These simulators were used by mineral engineers as an aid in plant design and process optimization and have since evolved into general purpose steady state simulation packages (Flintoff et al., 1985). Sastry and Adel (1984), who surveyed the state of simulation software in the mineral industry in 1984, concluded that most work has been in the areas of material balance packages and steady state flowsheet solvers, with little activity directed to dynamic simulation. The dynamic simulation work that has been done is in general, specific to a given circuit configuration or unit operation. Examples of these types of simulators for grinding operations are discussed by Bascur et al. (1985), del Villar and Laplante (1985), Finlayson and Hulbert (1980), Rajamani and Herbst (1980), and Smith and Guerin (1980). It is interesting to note that most of the control applications discussed in Section 1.2.1 utilized some form of computer simulation as an aid in the control system design.

Few simulation packages have considered the entire concentration circuit or have considered al the auxilliary equipment such as pumps and hydrocylones, with the exception of the work of Rajamani and Herbst (1980) and Bascur et al. (1985). It should be noted that although DYNAMILL (Rajamani and Herbst, 1980) includes pumping and particle size separation, it is limited to three specific circuit configurations: open circuit, closed circuit with post-classification, and closed circuit with pre-classification, each with only a single ball mill. Control strategies for the Duval Sierrita grinding circuit were evaluated using DYNAMILL by Bascur et al. (1985) with success.

Dynamic simulators for mineral processing operations that are general in nature are few in number. Adel (1982) has created a general interactive dynamic simulation package that simulates crushing, screening, grinding, hydrocyclone, and flotation unit operations either alone or in almost any flowsheet configuration. The development of this simulator was complicated by the fact that it was necessary to first develop a framework of utility programs for tasks such as integration and user input-output. It should be noted that this package is of limited use in evaluating control applications, mainly due to the lack of appropriate software. However, this simulator does encompass the most complete set of unit operations required for the simulation of an entire concentration circuit, but modelling of slurry pumping is not included. Work by Wood and Flintoff (1982), Wong (1984), and Flintoff et al. (1985), is based on DYFLO (Franks, 1972), a collection of FORTRAN subroutines for dynamic simulation in the chemical industry, which includes general utility and controller subroutines. This software, although capable of handling only grinding circuits, is very useful for studying process control applications. The approach is similar to that used by Adel (1982) and Rajamani and Herbst (1980) in that each mineral unit operation has been modelled by a separate subroutine. However, unlike DYNAMILL (Rajamani and Herbst, 1980) which will simulate only a single ball mill in three specific grinding circuit configurations, the DYFLO based simulator (Flintoff et al., 1985; Wong, 1984) is capable of simulating almost any grinding circuit flowsheet configuration including circuits with more than one mill. Flintoff et al. (1985) suggest that additional unit operations could easily be incorporated in their simulator to expand its capabilities to include all the operations normally associated with a complete concentration circuit.

1.2.4 Advanced Control System Simulation Studies with Applications in the Mineral Processing Industry

Simulation applications involving advanced control techniques such as optimal control using the instrumental variable recursive identification technique have been discussed in relation to a bauxite digestion circuit by Barrett-Lennard and Blair (1980). These authors describe the control system design and present simulation results which show that control performance has been improved. The extended Kalman filter has been used by Herbst et al. (1980) to identify ore hardness disturbances, while Rajamani (1985) describes a simulation study of a ball mill using self tuning control. Using mill speed to control the product particle size and circulating load in a ball mill circuit has been investigated by Herbst et al. (1983). These authors utilized results from pilot scale testing as the basis for the simulation work.

1.2.5 Grinding Circuit Control and Simulation at the University of Alberta

Since 1982 (Wood and Flintoff, 1982), work has been underway at the University of Alberta in developing a general dynamic grinding circuit simulation package. This initial research presented material suggesting that DYFLO (Franks, 1972), was suitable choice for providing a basis for a mineral processing simulator. This lead to the development of a suite of FORTRAN subroutines, each modelling a particular unit operation required for simulating grinding circuits. These unit operations included milling (both rod and ball types), particle size separation using hydrocyclones, slurry pumping, and solids transport. Wong (1984) followed this up by applying this early version of the simulator to the Lake Dufault grinding circuit, successfully utilizing the simulator to study various single input-single output control schemes with details of the simulator given by Flintoff et al. (1985). This early work has provided the framework for the development of a more complete dynamic concentration circuit simulator.

1.3 Thesis Organization

This thesis begins with Chapter 1, an introduction to simulation and control in the mineral processing industry and provides a literature survey of recent work. Chapter 2 contains a detailed discussion of the control theory employed in the simulation studies in Chapters 5 and 6, while the mathematical models used as the basis of the mineral processing unit operations are presented in Chapter 3. Chapter 4 describes the software package, MINSIM, developed as part of this work. This chapter serves as both the user's and programmer's manual. The Lake Dufault grinding circuit simulator version of MINSIM is utilized in the simulation study presented in Chapter 5. The performance of several advanced multivariable control schemes are investigated. Chapter 6 is concerned with simulation of discrete control of an open circuit grinding operation. It should be noted that both Chapters 5 and 6 are meant to demonstrate the many features available to the user of MINSIM. Conclusions from this work are presented in Chapter 7.

2. Control Theory

2.1 Introduction

Recently, it has become evident that the advantages afforded by the unit operation approach to process control used in the chemical industry can also be successfully applied to the mineral processing industry. Although this area of research is still in its infancy, relatively advanced control schemes developed for chemical processes can be directly applied to mineral processes. The purpose of this chapter is to provide a knowledge of the control theory used in the simulation studies presented in chapter 5, and to complement the FORTRAN IV source code presented in the appendices.

The following section begins with a discussion of proportional (P), proportional plus integral (PI), and proportional plus integral plus derivative (PID) controller forms commonly employed for single variable feedback control of continuous systems. From this discussion, section 2.3 formulates the discrete version of the PID controller. Feedforward control is discussed in section 2.4 and is related to improving the disturbance rejection characteristics of a control system when combined with the feedback control schemes presented in the previous two sections. Because feedforward controllers are typically implemented as lead/lag units, section 2.5 discusses this type of controller in further detail.

Following from the basic theory of single variable control, more advanced multivariable control system concepts are introduced in section 2.6. Included in this section are discussions on multiloop control, time delay compensation, interaction analysis, and noninteracting control design techniques. Finally, this chapter concludes with a brief presentation on adaptive control strategies in section 2.7.

2.2 Proportional, Proportional Integral, and Proportional Integral Derivative Feedback Controllers

2.2.1 Feedback Control

A general definition of a feedback control system is one in which a direct measurement of a controlled variable is used to trim the value of a manipulated variable such that a desired value of the controlled variable is maintained (Stephanopoulos, 1984). Figure 2.1 is a block diagram for this type of control loop configuration for a single input single output (SISO) process. The tasks associated with the controller block (G_c) in Figure 2.1 were historically performed by a human operator prior to the availability of instrumentation. More recently, the operators have been all but replaced by automatic controllers In fact, the automatic controllers quite often outperform their human counterparts.

The definition given above implies that an error signal is used to drive feedback controllers. The error signal is

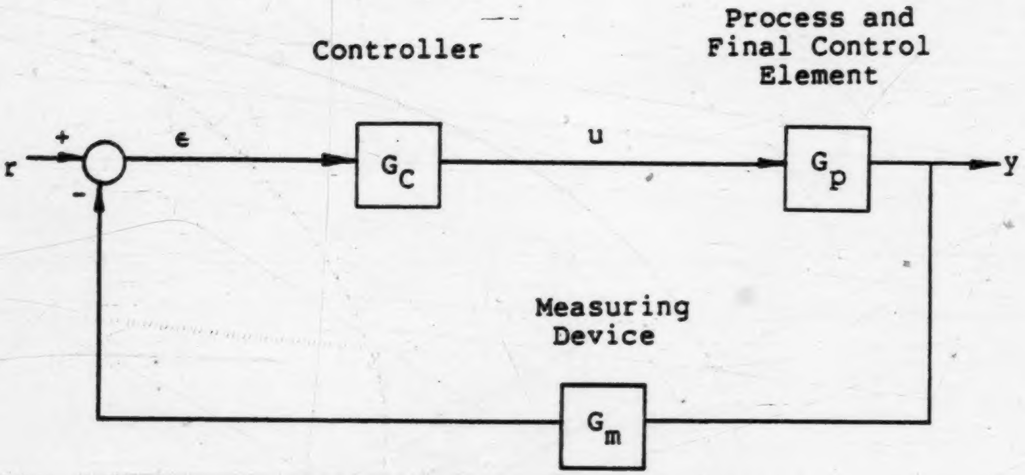

Controller
Process and Final Control Element
r
+
-
ε
G_c
u
G_p
y
Measuring Device
G_m

Figure 2.1: Block Diagram for a General SISO Feedback Control Loop Configuration

the difference between the desired and actual value of the controlled variable due either to a disturbance entering the process or a desired output setpoint change. This is described formally by the following equation, assuming $G_m=1$ and a negative feedback control loop configuration:

$$\epsilon(t) = r(t) - y(t) \tag{2.1}$$

where r(t) is the reference signal (setpoint) and y(t) is the controlled variable output value.

2.2.2 Proportional Control

The simplest automatic controller to make use of the error signal defined by equation (2.1) is the proportional (P) type controller. This controller produces an output signal proportional to the magnitude and sign of the error input signal and is represented by the following equation:

$$u_c(t) = K_c\epsilon(t) + g_b \tag{2.2}$$

where the term, g_b, is the controller bias and becomes the controller output when the error signal ($\epsilon(t)$) is zero. The controller output ($u_c(t)$) described by this equation will decrease as the value of the controlled variable increases above the setpoint because the error term as given in equation (2.1) becomes negative. Thus, equation (2.2) is written for a reverse acting controller. A direct acting controller (increase in controller output for an increase in the controlled variable above the setpoint) results when the controller gain has a negative value. For industrial controllers, negative gains are achieved by a reverse/direct action switch on the controller.

The transfer function for this type of controller is derived by first defining a deviation variable for the controller output variable as follows:

$$u_c'(t) = u_c(t) - g_b \tag{2.3}$$

and then rewriting equation (2.2) as:

$$u_c'(t) = K_c\epsilon(t) \tag{2.4}$$

It should be noted here that it is not necessary to introduce a deviation variable for the error because it is assumed to initially be zero. The transfer function follows from the Laplace transform of equation (2.4):

$$U_c(s) = K_cE(s) \tag{2.5}$$

Although this controller has the advantage that it is very simple and inexpensive, it is not suitable for many applications because a steady state error or controlled variable offset must exist to maintain control action. This can be explained by reconsidering equation (2.3). When an upset to the system occurs, be it a setpoint change or load, a relatively large error results and control action is taken. However, as steady state is approached, the error decreases and the controller output signal begins to return to its bias value. Clearly, this is unacceptable because some additional controller action is required to maintain the desired steady state and so, a steady state offset must exist to create an error signal. It should be noted here that increasing the proportional gain will alleviate this problem somewhat, however, the offset can never be completely removed without causing an unstable closed loop system response. The response curves shown in Figure 2.2 demonstrate this for a typical system under proportional control.

2.2.3 Proportional Integral Control

Because of the offset problem with the proportional controller, it is often desirable to add an integral mode to the control action as follows:

$$u_c(t) = K_c\left[\epsilon(t) + \frac{1}{\tau_I}\int\epsilon(t)dt\right] + g_b \tag{2.6}$$

The integral time constant, τ_I, in this equation determines

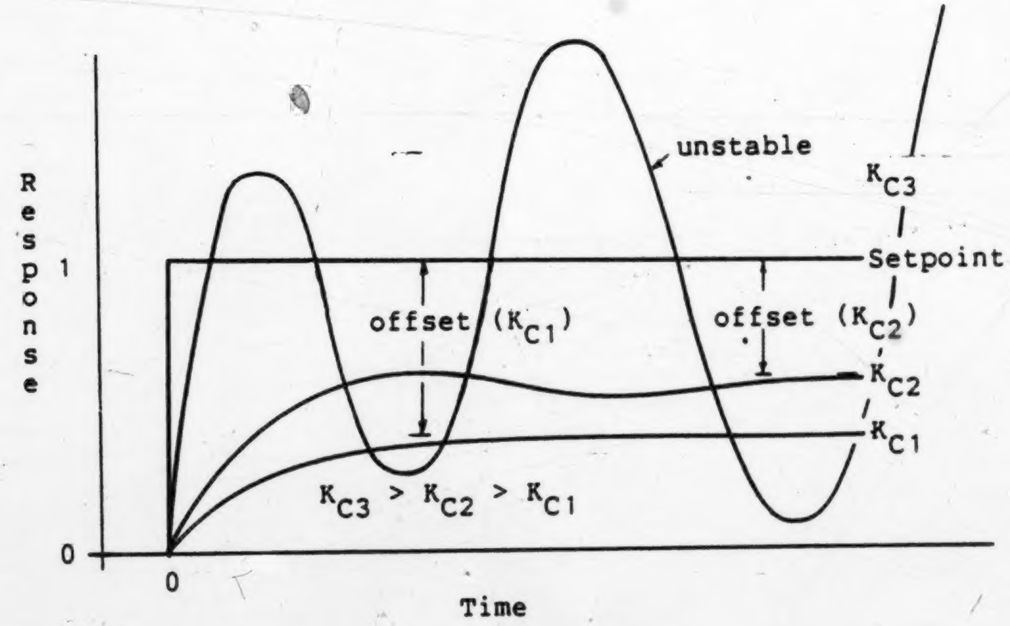


Figure 2.2: Typical Closed Loop System Response to a Unit Step Change in Setpoint Showing the Effect of Increased Proportional Gain on Stability and Offset

how fast the proportional action of the controller is repeated. This is further demonstrated in Figure 2.3 which presents the response of a PI controller for a unit step increase in the error. From this figure, it can be seen that a smaller time constant will produce a faster controller response.

The controlled variable offset is removed by this controller because the integral term operates as a sum. When an error in the controlled variable is present,

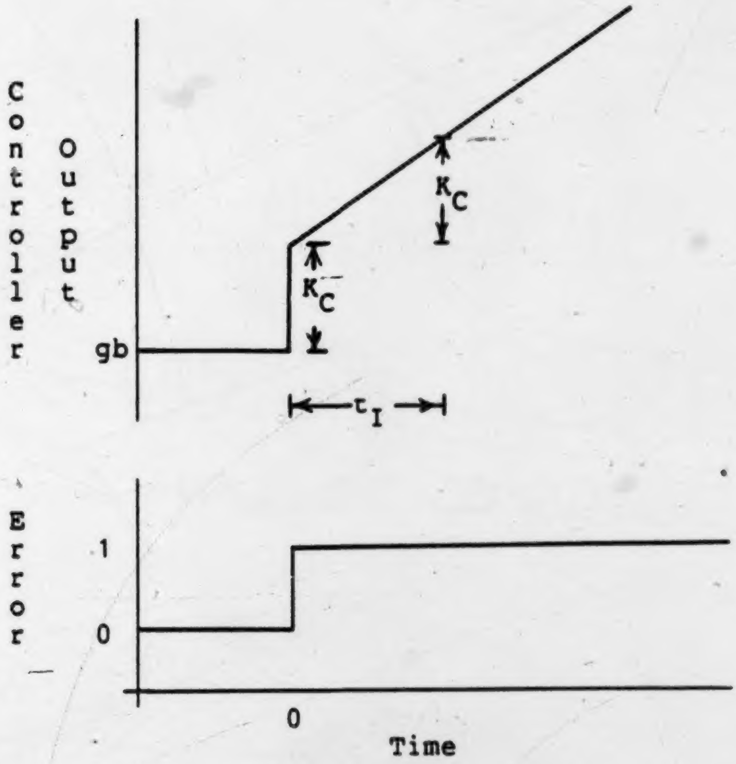


Figure 2.3: Response of a PI Controller To a Unit Step in the Input Error Signal

its value is integrated and added to the controller bias (g_b) and proportional action to form the controller output. As the system response approaches steady state, the error and proportional action both approach zero. However, the integral contribution to the controller output signal has eliminated any offset even though the error value is zero at steady state. In this case, equation (2.6) becomes:

$$u_c(t) = \frac{1}{\tau_I}\int 0dt + g_b \tag{2.7}$$

The integral term retains its value, effectively eliminating any steady state error from the controlled variable. It should also be pointed out that the integral term in this type of controller is often initialized to the value of the controller bias. This simplifies the construction of the controller and has no effect on the control action.

The transfer function representation for the PI controller can be developed in a fashion that closely follows the method used for the proportional controller in section 2.2.2. Using equation (2.1) with equation (2.6) and taking the Laplace transform yields:

$$U_c(s) = K_cE(s)\left[1 + \frac{1}{\tau_I s}\right] \tag{2.8}$$

Figure 2.4 shows the implementation of this controller in a block diagram format.

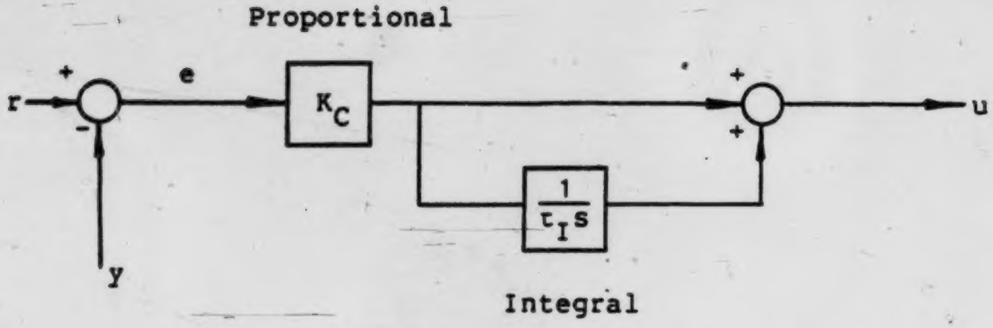


Figure 2.4: Block Diagram for an Ideal PI Controller

It should be noted here that integral control action has some disadvantages associated with it. In a closed loop situation, integral action will increase the order of the overall system through the addition of a lag into the characteristic polynomial, thus slowing down the system response. This problem can be alleviated by reducing the integral time constant or increasing the proportional gain, however, either of these solutions are capable of producing an unstable response to a system disturbance.

A second problem with PI controllers occurs when relatively large errors are present for extended periods of time such as during plant startup or cases where the system dynamics produce slow responses to load or setpoint disturbances. The integral action will continue to integrate and attempt to add to the controller output even after the final control element has reached its maximum (or minimum) position because an error still exists. This condition is commonly known as controller saturation or reset windup and will persist until the error signal changes sign so that the integral action contribution to the controller output begins to decrease. The most common method used to correct this problem is to limit the controller output to the operating range of the final control element and use a reset feedback structure implementation of the integral action. This structure follows from separating equation (2.8) into its proportional and integral components:

$$U(s)/E(s) = G_c(s) = K_c + G_I(s) \tag{2.9}$$

where:

$$G_I(s) = K_cE(s)\frac{1}{\tau_I s} \tag{2.10}$$

Combining equation (2.10) with the solution for $K_cE(s)$ from equation (2.9) yields the integral action term:

$$G_I(s) = \left[\frac{1}{\tau_I s + 1}\right]G_c(s) \tag{2.11}$$

The block diagram for the implementation of the controller defined by equations (2.9) and (2.11) is shown in Figure 2.5. A typical pneumatic limiter is shown in this figure and prevents the controller output from exceeding the range

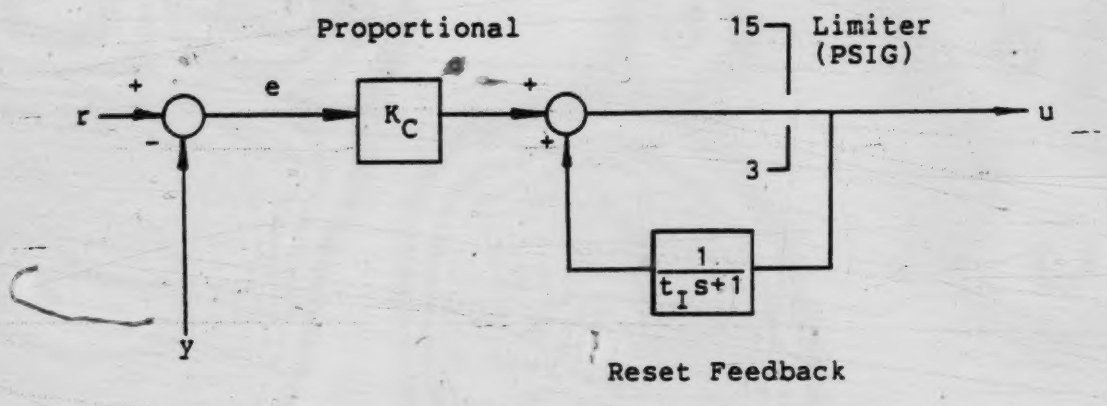


Figure 2.5: Block Diagram Showing Reset Feedback Implementation of a PI Controller with an Output Limiter for Reset Windup Protection

allowed by the final control element. The limited controller output signal is fed back to the integral part of the controller. The gain of this element of the controller is unity and its output always lags behind the controller output. Thus, reset windup protection is automatic with this PI controller configuration.

2.2.4 Proportional Integral Derivative Control

Derivative action can be added to the PI controller algorithm presented in section 2.2.3 to further improve the overall control system performance. The objective of this controller mode is to anticipate the direction in which the output variable is proceeding, and its rate of change. This controller is represented by the following equation:

$$u_c(t) = K_c\left[\epsilon(t)+\frac{1}{\tau_I}\int\epsilon(t)dt+\tau_D\frac{d\epsilon(t)}{dt}\right]+g_b \qquad (2.12)$$

The addition of derivative action to the controller equation has the effect of slowing the system response because it increases the effective system time constant as did the addition of integral action. The reduction in response speed comes because the overall system damping is increased by an amount proportional to the derivative time constant, τ_D. This stabilizing effect of the derivative action improves the robustness of the controlled system and allows the proportional gain, K_c, to be increased to obtain an acceptable response speed. The oscillatory response associated with an increase in this gain is tempered by the anticipatory nature of the derivative action.

The transfer function representation of the PID controller can again be developed in a fashion similar to the method used for the proportional controller in section 2.2.2. Using equation (2.1) with equation (2.12), and taking the Laplace transform results in:

$$U_c(s) = K_cE(s)\left[1 + \frac{1}{\tau_I s} + \tau_D s\right] \qquad (2.13)$$

which is the ideal continuous PID controller transfer function. Analog implementation of this ideal transfer function is not feasible because the derivative calculation is not possible. A lead/lag network is usually used to approximate the derivative term resulting in the actual transfer function:

$$U_C(s) = K_C E(s)\left[1 + \frac{1}{\tau_I s}\right]\left[\frac{\tau_D s+1}{a\tau_D s+1}\right] \qquad (2.14)$$

The parameter, *a*, is called the noise filter parameter or rate amplitude and its value is set by the manufacturer and typically ranges between 0.05 and 0.10 (Smith and Corripio, 1985). Clearly, a smaller rate amplitude results in a closer approximation of the ideal transfer function.

Another problem with PID controllers is the fact that derivative action is extremely sensitive and may produce an undesirably large control action or derivative "kick" for a small error input signal. This implies that derivative action should not be used when measurement or process noise exist. Setpoint changes will cause a controller "kick" as well, however, this latter problem can be alleviated by using a PID structure which involves derivative action on the output signal. Implementation of this structure is shown in the block diagram in Figure 2.6. It should be noted that the derivative action in this controller acts as a filter on the output signal before the error calculation is performed. The remaining elements of this controller take the form of the PI controller discussed in section

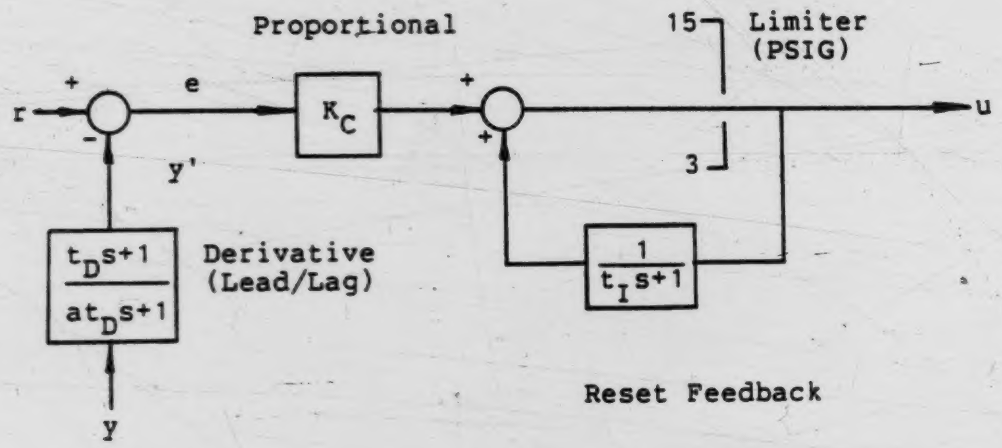

Proportional
15
Limiter
(PSIG)
+
e
r
-
y'
K_C
+
+
u
3
$\frac{t_D s+1}{at_D s+1}$
Derivative
(Lead/Lag)
$\frac{1}{t_I s+1}$
Reset Feedback
y

Figure 2.6: Block Diagram for a PID Controller with Derivative Action Only on the Output Signal Using a Lead Lag Approximation of the Ideal Derivative Action

2.2.3.

2.3 Discrete PID Controllers

2.3.1 Discrete PID Algorithms

Starting from the ideal, continuous time domain representation of a PID controller given below:

$$u(t) = K_c \left| \epsilon(t) + \frac{1}{\tau_I} \int \epsilon(t)dt + \tau_D \frac{d\epsilon(t)}{dt} \right| \qquad (2.15)$$

where:

$$\epsilon(t) = y_{sp}(t) - y(t) \qquad (2.16)$$

it follows that a difference equation will result if equation (2.15) is discretized. The most straight forward procedure is to approximate the integral term with a summation, and the derivative term with a difference equation. Using the trapezoidal integration rule and first order backward differencing for the derivative term yields:

$$u(k) = K_c\left[\epsilon(k) + \frac{T_s}{\tau_I}\left[\frac{\epsilon(0)+\epsilon(k)}{2} + \sum_{i=1}^{k-1} \epsilon(i)\right]\right.$$

$$\left. + \frac{\tau_D}{T_s}[\epsilon(k) - \epsilon(k-1)]\right] \tag{2.17}$$

where T_s is the sample interval. Equation (2.17) is the positional form of the classical, discrete PID algorithm because the absolute position of the final control element is calculated. For computer applications, a recursive form, known as the incremental or velocity form, is more appropriate.

For a change in the value of the manipulated variable from time k-1 to time k, that is:

$$\Delta u = u(k) - u(k-1) \tag{2.18}$$

it follows from equation (2.17) that the increment in the controller output is given by:

$$\Delta u = K_C\left[\epsilon(k)-\epsilon(k-1) + \frac{T_s}{\tau_I}\left[\frac{\epsilon(k)+\epsilon(k-1)}{2}\right] + \frac{\tau_D}{T_s}[\epsilon(k)-2\epsilon(k-1)+\epsilon(k-2)]\right] \tag{2.19}$$

Substitution for Δu in equation (2.19) using equation (2.18) gives:

$$u(k) = u(k-1) + K_C\left[\epsilon(k)-\epsilon(k-1) + \frac{T_s}{\tau_I}\left[\frac{\epsilon(k)+\epsilon(k-1)}{2}\right] + \frac{\tau_D}{T_s}[\epsilon(k)-2\epsilon(k-1)+\epsilon(k-2)]\right] \tag{2.20}$$

which is commonly known as the velocity form of the discrete PID algorithm. This form of the discrete PID controller has further advantages over the positional form given in equation (2.17) other than being more appropriate for computer calculations. By virtue of the velocity form, bumpless transfer from manual to automatic control is obtained without initialization of the integrator and anti-reset windup protection is inherent when the controlled variable is limited.

An additional benefit can be gained by writing equation (2.20) in terms of conventional PID controller constants:

$$u(k) = u(k-1) + K_p[\epsilon(k)-\epsilon(k-1)] + K_I\left[\frac{\epsilon(k)+\epsilon(k-1)}{2}\right]T_s + \frac{K_D}{T_s}[\epsilon(k)-2\epsilon(k-1)+\epsilon(k-2)] \qquad (2.21)$$

where:

$K_p = K_c$

$K_I = K_c/\tau_I$

$K_D = K_c*\tau_D$

This algorithm has the advantage that the controller settings are noninteracting.

2.3.2 Discrete PID Structures

The ideal PID structure has been demonstrated in all the preceeding algorithms and its representation in a feedback loop is presented in Figure 2.7. Although this classical form is commonly seen in textbooks and is in generally suitable for analytical studies, various other structures exist with more desirable features that more closely approximate actual controllers.

The most useful structural change is to remove the derivative action from the error signal and place it on the output variable signal only. This derivative action acting

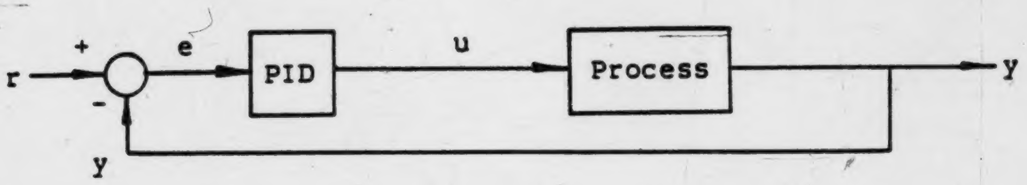


Figure 2.7: Block Diagram a the Classical Discrete PID Controller

only on the output signal is shown in the block diagram in Figure 2.8 and has the advantage that setpoint changes will not cause a large derivative "kick" in the controller output. The PID algorithm to incorporate this feature is easily developed by eliminating the setpoint from the error calculation (c.f. equation (2.16)) in the derivative term of equation (2.21) as follows:

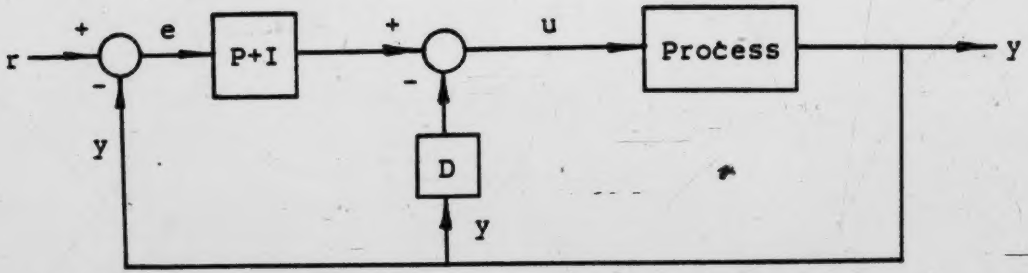


Figure 2.8: Block Diagram for a Discrete PID Controller with Derivative Action Only on the Output Signal

$$u(k) = u(k-1) + K_p[\epsilon(k)-\epsilon(k-1)] + K_I\left[\frac{\epsilon(k)+\epsilon(k-1)}{2}\right]T_s + \frac{K_D}{T_s}[-y(k)+2y(k-1)-y(k-2)] \qquad (2.22)$$

It should be noted that equation (2.22) retains the velocity form of equation (2.21) so is a commonly used discrete PID algorithm.

A further modification to the classical structure follows from the above discussion, and results in the integral action only on the setpoint structure as shown in the block diagram in Figure 2.9. The appropriate algorithm is:

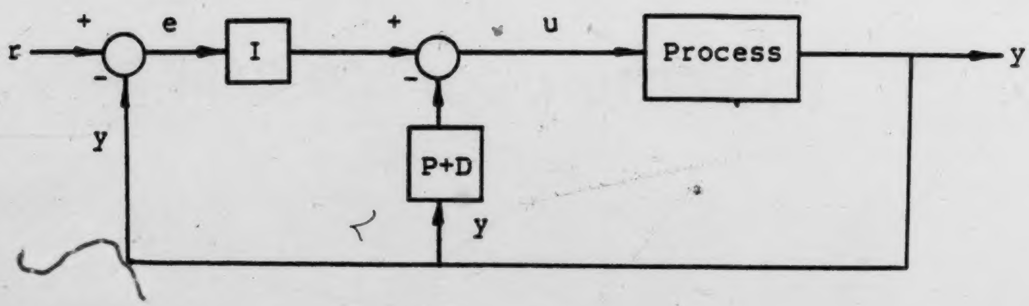


Figure 2.9: Block Diagram for a Discrete PID Controller with Integral Action Only on the Setpoint Signal

$$u(k) = u(k-1) + K_p[-y(k)+y(k-1)] + K_I\left[\frac{\epsilon(k)+\epsilon(k-1)}{2}\right]T_s + \frac{K_D}{T_s}[-y(k)+2y(k-1)-y(k-2)] \tag{2.23}$$

It can be seen that the setpoint in the error calcuation has been removed from both the derivative and proportional terms of this algorithm to avoid proportional as well as derivative "kicks". The disadvantage of this form is that some of the generality is lost. The derivative action on only the output variable, shown in Figure 2.8, allows the controller to be operated with any combination of the three different types of action (except derivative action only) simply by setting the appropriate controller constants to zero. The structure in Figure 2.9 requires that integral action always be used and thus, this structure has limited practical application.

2.4 Feedforward Controllers

Feedforward controllers find application in systems where measurable disturbances are present and as the name implies, allow for control action to be taken before disturbances affect system behavior. This differs from the feedback concept in that the feedback scheme relies on a deviation from the setpoint (error) to drive the controller output after the disturbance has already upset the process. The feedforward scheme depends only on the ability to measure incoming disturbances. Because not all disturbances are measurable, and because it is difficult to determine G_{ff} exactly, it is normal practice to combine both feedforward and feedback schemes in a given control system. A block diagram for this type of control system is shown in Figure 2.10. It should be noted that the potential for perfect control theoretically exists with feedforward control as long as all disturbances can be identified (modelled) and measured accurately.

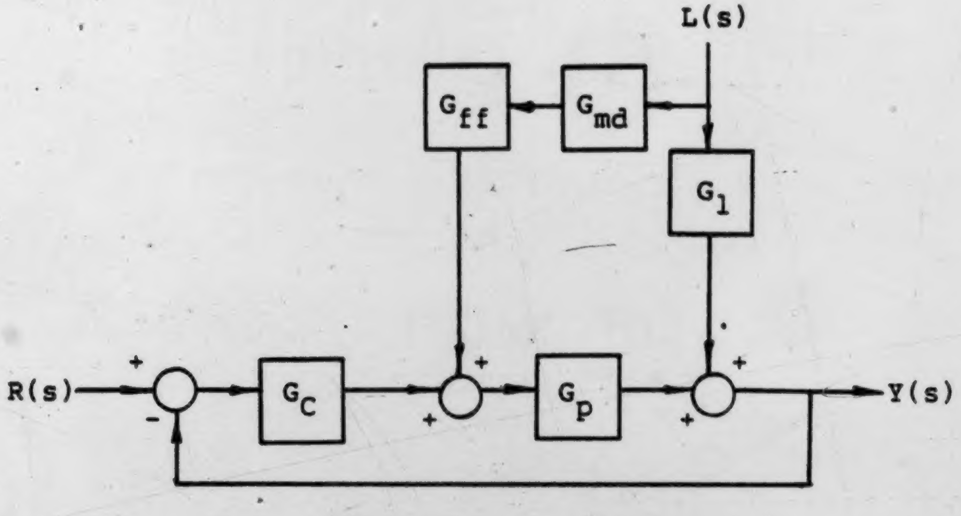


Figure 2.10: Block Diagram for a Feedforward - Feedback Control Scheme

The design of a feedforward controller follows from an analysis of the block diagram in Figure 2.10. The open loop transfer function for this block diagram is:

$$G_{ol} = G_c G_p \tag{2.24}$$

and it follows that the closed loop transfer function may be writen as:

$$Y(s) = \left[\frac{G_c G_p}{1+G_{ol}}\right] R(s) + \left[\frac{G_l}{1+G_{ol}}\right] L(s) + \left[\frac{G_{md} G_{ff} G_p}{1+G_{ol}}\right] L(s) \tag{2.25}$$

It can be seen from equation (2.25) that the addition of feedforward control does not affect the overall control system stability because the feedforward controller does not appear in the characteristic polynomial.

To reject disturbances, the contribution of the latter two terms in equation (2.25) must combine to give zero. This can be accomplished by specifying a feedforward controller (G_{ff}) such that:

$$0 = G_l + G_{md} G_{ff} G_p \tag{2.26}$$

and so, it follows that:

$$G_{ff} = \frac{-G_1}{G_{md}G_p} \tag{2.27}$$

As the feedforward controller in equation (2.27) is composed of several different components of the overall control system, its form will vary depending on the form of the individual components. It is clear that a very complex controller could result under certain circumstances. Fortunately, some simplifying assumptions can usually be made.

A static feedforward controller design will result from consideration of only the steady state gain of each component:

$$G_{ff} = \frac{-K_1}{K_{md}K_p} \tag{2.28}$$

This controller is adequate if the dynamics of the load and process transfer functions are approximately the same and if the disturbance measurement device dynamics can be neglected. This controller can be implemented simply by use of a gain-only device. Usually the dynamics of the load and process transfer functions differ significantly and so a more complex controller of the form:

$$G_{ff} = \frac{-G_1}{K_{md}G_p} \tag{2.29}$$

is required. It should be noted that analog implementation of the controller defined by equation (2.29) may not be economically feasible if the difference in order of either the load and process transfer function is greater than 2 or if a time delay term results. However, if a digital computer is being utilized, these problems are of no concern. Physical realizability will depend only on the realizability of time delay terms (i.e. "predictive" time delays are not realizable).

Commonly, the load and process transfer functions may be adequately characterized by first order lag models with the appropriate gains cascaded with time delays. This will result in a feedforward controller of the form:

$$G_{ff} = \frac{-K(\tau_1 s+1)e^{-T_d s}}{\tau_2 s+1} \tag{2.30}$$

which is simply a first order lead/lag unit, usually available "off the shelf". As before, realizability depends on the realizability of the time delay term.

It should be noted that all of the above forms for feedforward controllers are sensitive to process and disturbance modelling errors. Clearly, parameter variations in either model will affect the accuracy of the controller design and thus, the performance of the feedforward control action. This particular problem demonstrates another reason for using feedback control in conjunction with a feedforward scheme. The feedback controller is able to compensate for any slight inadequacies in the feedforward controller performance.

2.5 Lead-Lag Compensators

In view of the common lead-lag form of feedforward controllers, it is pertinent to discuss lead-lag networks. A lead-lag unit may be described by the following transfer function:

$$G(s) = \frac{\tau_1 s+1}{\tau_2 s+1} \tag{2.31}$$

where:

τ_1 is the lead time constant

τ_2 is the lag time constant

To develop a DYFLO2-like FORTRAN subroutine, a time domain representation of equation (2.31) is needed. The most obvious procedure is to simply form the differential equation that resulted in equation (2.31) when the Laplace transformation is taken:

$$\frac{dy(t)}{dt} = \frac{1}{\tau_2}\left[\tau_1\frac{dx(t)}{dt}+x(t)-y(t) \right] \tag{2.32}$$

where:

y(t) is the output signal

x(t) is the input signal

Clearly, this equation is not without disadvantages, particularly the derivative of the input signal term on the righthand side. Implementation of this compensator on a computer will require that this differentiation be carried out numerically. If the input signal contains any noise at all, wild derivative behaviour can be expected!

To avoid this problem, a less direct approach which is presented in the following material, as suggested by Smith and Corripio (1985), can be employed.

From the original Laplace transform given in equation (2.31), let:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{\tau_1 s+1}{\tau_2 s+1} \tag{2.33}$$

Then, rearranging gives:

$$\tau_2 sY(s)+Y(s) = \tau_1 sX(s)+X(s) \tag{2.34}$$

Collecting the terms of s and dividing by τ_2 gives:

$$s\left[Y(s)-\frac{\tau_1}{\tau_2}X(s) \right] = \frac{1}{\tau_2}\left[X(s)-Y(s) \right] \tag{2.35}$$

Defining a new (artificial) state variable as:

$$Y_1(s) = Y(s) - \frac{\tau_1}{\tau_2}X(s) \tag{2.36}$$

and then substituting (2.36) into (2.35) yields:

$$sY_1(s) = \frac{1}{\tau_2}\left[X(s)-Y(s) \right] \tag{2.37}$$

Inverting the Laplace transform in equation (2.37), it follows that:

$$\frac{dy_1(t)}{dt} = \frac{1}{\tau_2}[\ x(t)-y(t)\ ] \tag{2.38}$$

Integration of equation (2.38) gives $y_1(t)$. Substituting this result into the inverse transformation of equation (2.36) gives the actual output, y(t) as follows:

$$y(t) = y_1(t) + \frac{\tau_1}{\tau_2}x(t) \tag{2.39}$$

The lead-lag network is computed using equations (2.38) and (2.39) and the compensator is completed by multiplying the output given by equation (2.39) by a gain term as follows:

$$y_a(t) = Ky(t) \tag{2.40}$$

2.6 Control of a Multivariable System

In general, the single input single output (SISO) type systems discussed thus far are an exception, rather than the rule, in most real processes. Usually, the physical nature of the process dictates a multiple input multiple output (MIMO) configuration with n input (manipulated) and m output (controlled) variables as shown in Figure 2.11. Unlike the SISO case where one input affects only one output, interaction between the various inputs and outputs is almost

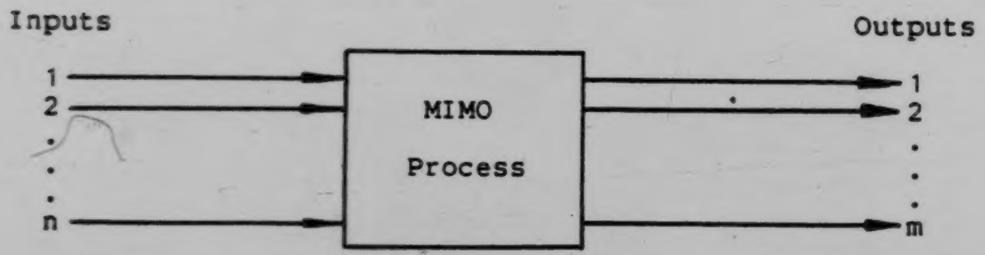
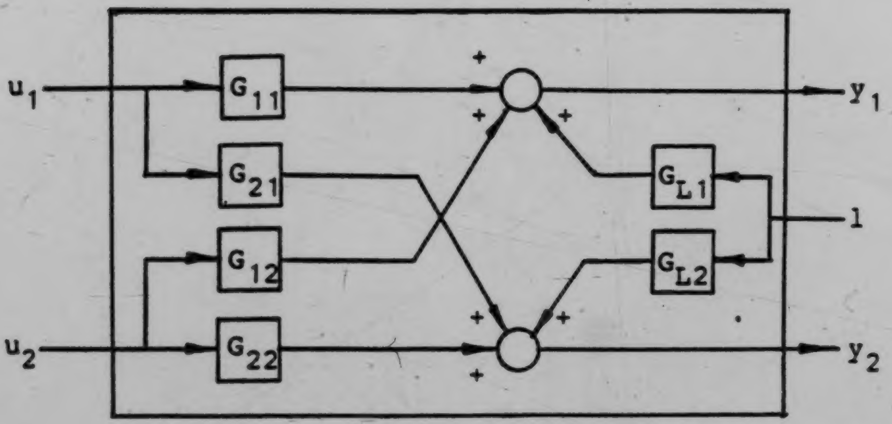


Figure 2.11: A General Multivariable Process

always present in MIMO processes. Figure 2.12 presents a block diagram for a general two by two interacting process. Included in this figure is a load variable input which may be either measurable or unmeasurable. Clearly, more than one load variable can be present in a given process. This figure implies that the number of inputs to a process does not necessarily match the number of outputs (i.e. m ≠ n).

For this general two by two case, the process transfer function matrix is defined as:



MIMO Process

Figure 2.12: A General Two by Two Interacting Process

$$G_p = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \qquad (2.41)$$

and the load transfer function vector is:

$$G_L = \begin{bmatrix} G_{L1} \\ G_{L2} \end{bmatrix}$$

Each element in the above two equations corresponds to the appropriate block in Figure 2.12.

2.6.1 Multiloop Control

To control a process such as that presented in Figure 2.12 it is not uncommon practice to ignore the interactions and design a control system using SISO techniques based only on the direct transmission transfer functions, G_{11} and G_{22}. The configuration of this control scheme is shown in Figure 2.13. For convenience of analysis, an equivalent multivariable block diagram is presented in Figure 2.14 in terms of matrix notation. Using this latter block diagram, the closed loop transfer function can be written in matrix notation as:

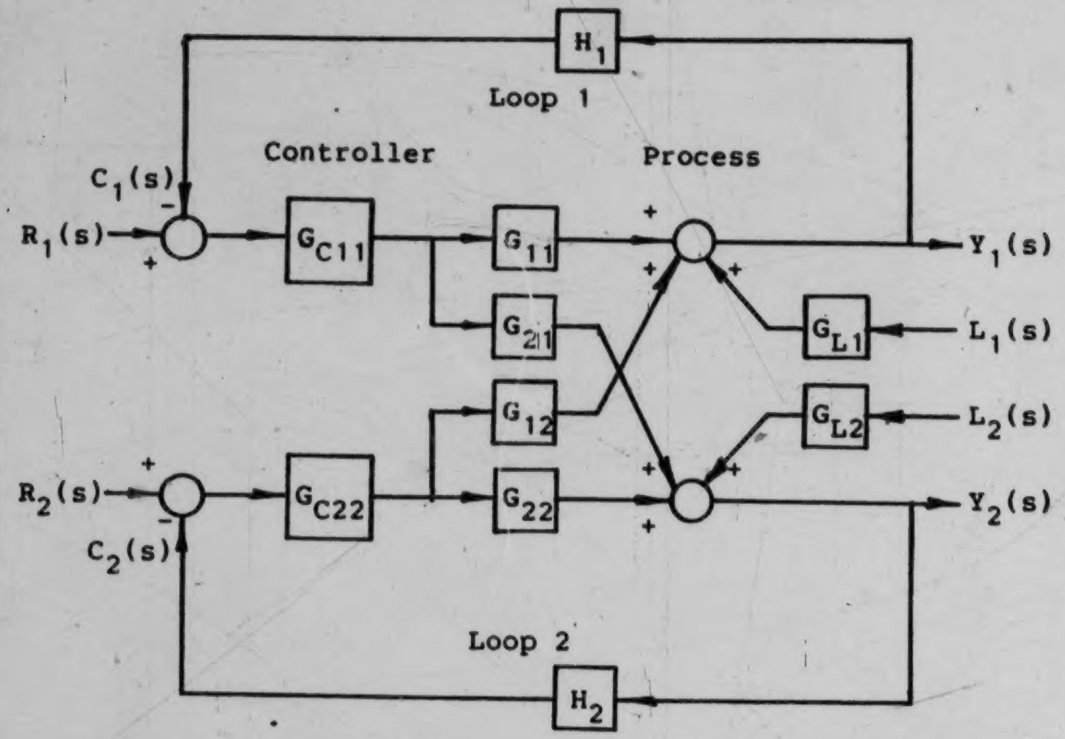

Loop 1
Controller
Process
C1(s)
R1(s)
GC11
G11
G21
G12
G22
GC22
R2(s)
C2(s)
GL1
GL2
Y1(s)
L1(s)
L2(s)
Y2(s)
H1
H2
Loop 2

Figure 2.13: A Classical Multiloop Control System Ignoring Process Interactions

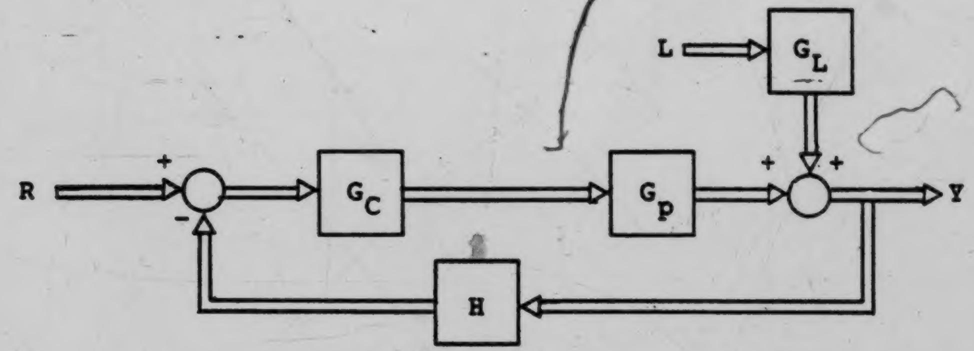


Figure 2.14: Equivalent Representation of a Block Diagram for a Multiloop Control System

$$Y = [I + G_pG_cH]^{-1}[G_pG_cR + G_LL] \tag{2.42}$$

where:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}; \; I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \; G_p = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix};$$

$$H = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}; \; G_c = \begin{bmatrix} G_{c11} & 0 \\ 0 & G_{c22} \end{bmatrix}; \; R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

$$G_L = \begin{bmatrix} G_{L1} \\ G_{L2} \end{bmatrix}; \; L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}$$

This equation can be rewritten as:

$$Y = PR + QL \tag{2.43}$$

where:

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \tag{2.44}$$

and:

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{2.45}$$

Performing the appropriate matrix algebra, the individual terms in the above two matrices are found to be:

$$P_{11} = \frac{G_{11}G_{C11}+G_{22}G_{C22}H_2G_{11}G_{C11}-G_{12}G_{C22}H_2G_{21}G_{C11}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 -G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \tag{2.46}$$

$$P_{12} = \frac{G_{21}G_{C11}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 -G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \tag{2.47}$$

$$P_{21} = \frac{G_{12}G_{C22}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 -G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \tag{2.48}$$

$$P_{22} = \frac{G_{22}G_{C22}+G_{11}G_{C11}H_1G_{22}G_{C22}-G_{21}G_{C11}H_1G_{12}G_{C22}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 -G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \tag{2.49}$$

and:

$$Q_1 = \frac{G_{L1}+G_{22}G_{C22}H_2G_{L1}-G_{12}G_{C22}H_2G_{L2}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 - G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \qquad (2.50)$$

$$Q_2 = \frac{G_{L2}+G_{11}G_{C11}H_1G_{L2}-G_{21}G_{C11}H_1G_{L1}}{1+G_{22}G_{C22}H_2+G_{11}G_{C11}H_1+G_{11}G_{C11}H_1G_{22}G_{C22}H_2 - G_{12}G_{C22}H_2G_{21}G_{C11}H_1} \qquad (2.51)$$

By writing the transfer functions for each loop as:

$$Y_1 = P_{11}R_1 + P_{12}R_2 + Q_1L_1 \qquad (2.52)$$

and:

$$Y_2 = P_{21}R_1 + P_{22}R_2 + Q_2L_2 \qquad (2.53)$$

it is easily seen that an upset in loop 1 will affect Y_2 and an upset in loop 2 will affect Y_1. Depending on the strength of the interaction, one or both of the controllers may require detuning before satisfactory controlled response behavior will result. This behavior naturally leads to a consideration of the extent of the interaction and in the presence of interaction what strategy can be adopted to improve the behavior of the controlled system. These matters are considered in the following two sections.

2.6.2 Interaction Analysis

In most cases an optimum choice for the pairing of manipulated (input) and controlled (output) variables exists that will result in minimum interaction between the various possible control loops. Several interaction measures and indices have been posed over the last three decades. One of the first methods, called the relative gain array (RGA) technique, developed by Bristol (1966) is widely used because it is easy to understand, simple to use, and provides a good insight into expected control behavior. It uses steady state gains to suggest appropriate variable pairings and does not consider possible effects of the process dynamics. Witcher and McAvoy (1977) have extended the ideas of the RGA to the dynamic case and Tung and Edgar (1981) have done an analysis of interaction in both the frequency and time domains. Although this method provides more information and may in fact suggest opposite pairing compared with the result from use of the RGA analysis, it is more difficult to use because more computational effort is required. Hence, following the original objectives of this thesis (c.f. Chapter 1), only the steady state relative gain array will be discussed in detail. It should be noted that other steady state interaction indices and extensions have been suggested by other researchers.

The relative gain array, for the general two by two system shown in Figure 2.12 is defined as:

$$\text{RGA} = \begin{array}{c|cc} & u_1 & u_2 \\ \hline y_1 & \lambda_{11} & \lambda_{12} \\ y_2 & \lambda_{21} & \lambda_{22} \end{array}$$

~~(2.54)~~

where:

$$\lambda_{ij} = \frac{(\delta y_i/\delta u_j)_{\bar{u}}}{(\delta y_i/\delta u_j)_{\bar{y}}} \tag{2.55}$$

Each element of the array in equation (2.54), as defined by equation (2.55), is a ratio of the gain of output i with respect to input j when all other loops are open and the gain of output i with respect to input j when all other loops are under closed loop control. Thus, the relative gain array is in effect a normalized measure of how much each manipulated variable affects each controlled variable when all other loops are closed.

The numerator terms in equation (2.55) come from the open loop steady state process gain matrix, which is defined as follows:

$$A = \begin{bmatrix} (\delta y_1/\delta u_1)_{\bar{u}} & (\delta y_1/\delta u_2)_{\bar{u}} \\ (\delta y_2/\delta u_1)_{\bar{u}} & (\delta y_2/\delta u_2)_{\bar{u}} \end{bmatrix} \tag{2.56}$$

The closed loop counterpart or denominator in equation

(2.55) can be calculated from:

$$B = \begin{bmatrix} (\delta u_1/\delta y_1)_{\bar{y}} & (\delta u_2/\delta y_1)_{\bar{y}} \\ (\delta u_1/\delta y_2)_{\bar{y}} & (\delta u_2/\delta y_2)_{\bar{y}} \end{bmatrix} \tag{2.57}$$

It should be noted however, that only one of the above two equations must be explicitly calculated because equations (2.56) and(2.57) are related through the equation:

$$B = (A^{-1})^T \tag{2.58}$$

The elements of the relative gain array can be calculated from equation (2.55) using the appropriate elements of the equations (2.56) and (2.57), noting that the reciprocals of each individual term in equation (2.57) are used in equation (2.55).

This discussion has assumed that a model of the plant is available for use in evaluating the partial derivatives in equations (2.56) and (2.57), either analytically or numerically. If an appropriate model of an actual plant does not exist, A can be approximated from experimental open loop response testing. In other words, the partial derivatives in equation (2.56) become:

$$A = \begin{bmatrix} (\Delta y_1/\Delta u_1)_{\bar{u}} & (\Delta y_1/\Delta u_2)_{\bar{u}} \\ (\Delta y_2/\Delta u_1)_{\bar{u}} & (\Delta y_2/\Delta u_2)_{\bar{u}} \end{bmatrix} \tag{2.59}$$

and B is calculated using equation (2.57). The absence of load disturbances to the system is essential for the success of this procedure.

Before proceeding to control loop selection, it is appropriate to consider the significance of the numerical values of the elements of the relative gain array. The significance of the values can be summarized as follows:

(i) $\lambda_{ij} = 1$ indicates no interaction between the i^{th} controlled variable and the j^{th} manipulated variable or that interaction of this loop with all others is offsetting (i.e. the ij^{th} elements of equations (2.56) and (2.57) are reciprocals of each other). In either case, the loop appears to be completely decoupled from the others. This pairing is favorable.

(ii) $\lambda_{ij} = 0$ indicates that the j^{th} manipulated variable has no effect on the i^{th} controlled variable either because the steady state open loop gain is zero, or very strong interactions with other closed loops. This pairing should not be used because the system is uncontrollable.

(iii) $0 < \lambda_{ij} < 1$ indicates a varying degree of positive interaction among the various controlled and manipulated variables. In general, the smaller the value, the stronger the interaction.

(iv) $\lambda_{ij} < 0$ indicates a negative interaction where the j^{th} manipulated variable causes the controlled variable to move in the opposite direction when the other loops are closed (i.e. the closed loop gain changes sign). The controller action depends on the mode (automatic or manual) of the other loops in the system. Clearly, this type of pairing is undesirable and even dangerous!

(v) $\lambda_{ij} > 1$ indicates a negative interaction. However, the sign of the closed loop gain does not change when the other loops are closed as when $\lambda_{ij} < 0$, but instead, the control actions will tend to oppose each other. The gain of this variable pairing is cut by a factor of $1/\lambda_{ij}$ when the other loops are closed and larger controller gains will be required.

It should be clear if the significance of these five points is understood that control loop interactions can be minimized by using an appropriate input output variable pairing based on the values of the elements of the relative gain array. A general variable pairing rule for this technique can be stated as follows:

> Pair only controlled and manipulated variables that correspond to relative gain matrix elements that are both positive and have numerical values as close to unity as possible.

It is interesting to note that the latter three points in the above list bring out the fact that there are two possible kinds of interaction: positive and negative. Positive interaction is the most desirable form and occurs when the process has an odd number of positive (or negative) steady state gains. Controlling this type of process is eased by the fact that the respective control actions tend to compliment each other. Negative interaction occurs when there are even numbers of positive and negative steady state process gains, or when the gains are all of the same sign. In this case, as mentioned earlier, the proper input-output variable pairing results in control loops with action that opposes each other. Appendix E presents an interaction analysis example for a negative interaction process.

Another interesting and useful property of the relative gain array is that the rows and columns must always sum to unity. The implication of this is that for the two by two system described in this section, only one element of the relative gain array needs to be explicitly determined using equation (2.55). The remaining three elements can be found using the aforementioned property. For a general nxn RGA, the minimum number of elements that require explicit calculation can be found using the formula:

$$\text{Number} = (n-1)^2 \tag{2.60}$$

The final point to be covered in this discussion is with respect to an unequal number of process inputs and outputs. As the relative gain array is square, it is implied that the number of inputs must equal the number of outputs. However, given an unequal number, several unique relative gain arrays can be formed for each unique set of possible variable pairings. The number of relative gain arrays required clearly depends on the number of unique pairings and all require analysis before the minimal interaction choice can be made.

2.6.3 Decoupling Controllers

The next logical progression in mulivariable controller design, after the interaction analysis has been performed, is to attempt to eliminate the unfavorable couplings. This can be done with the aid of decoupling controllers.

Recently, much work has gone into decoupling controller design techniques. The basic purpose in all cases is to at least minimize the "off diagonal" terms in the closed loop transfer function matrix (TFM). Wood (1977) gives an overview of the more commonly used multivariable design techniques such as the characteristic loci, inverse Nyquist array, and the direct Nyquist array. The common objective of all of these methods is to minimize interaction between the loops. These methods, extensions of the classical single variable frequency domain design techniques to multivariable systems, are iterative in nature and the computer aided design (CAD) systems are required to use the techniques efficiently (Wood, 1977). As the purpose of this work is to provide the plant engineer with quick solutions, only the simpler noninteracting/decoupling design techniques are of interest and usually prove to be quite adequate.

U

The noninteracting/decoupling techniques result in a completely noninteractive system (i.e. off diagonal elements in the closed loop transfer function matrix are identically zero) if the process is both well known and well behaved. Although these methods are a subset of the frequency domain techniques, they are based on steady state operating conditions, and as a consequence, the performance of the resulting control system may not be as satisfactory as for a system designed using the dynamic techniques. This, of course, will depend on the reliability of the process transfer function model used in the design. The advantage of steady state design however, is that a CAD system is not required and the individual elements of the control system are usually more easily realized. The MIMO design problem is reduced by complete decoupling to a series of SISO problems on which conventional techniques can be used.

For simplicity, the theory will be developed for the two by two interacting system previously shown in the block diagram in Figure 2.12. Although no additional benefit could be gained by doing so here, it is a simple matter to extend this theory to an n by n system.

The basic concept for all noninteracting control schemes is the creation of other paths of interaction between the control loops in an attempt to cancel the original interactions due to G_{12} and G_{21} (c.f. equations (2.46) and (2.47)). The most common procedure is to place a decoupling precompensator on the process as shown in the block diagram in Figure 2.15. An equivalent representation of this block diagram is shown in Figure 2.16 and proves to be convenient for the analysis that follows. The closed loop transfer function for this block diagram, written in matrix notation, is:

$$Y = [I + G_p D G_C H]^{-1}[G_p D G_C R + G_L L] \tag{2.61}$$

where:

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

Equation (2.61) may be rewritten as:

$$Y = PR + QL \tag{2.62}$$

in a similar fashion as was done for the 'interacting case presented earlier. Again, performing the matrix operations indicated by equation (2.61) results in an expression for each individual element of the P and Q matrices in equation (2.62) as follows:

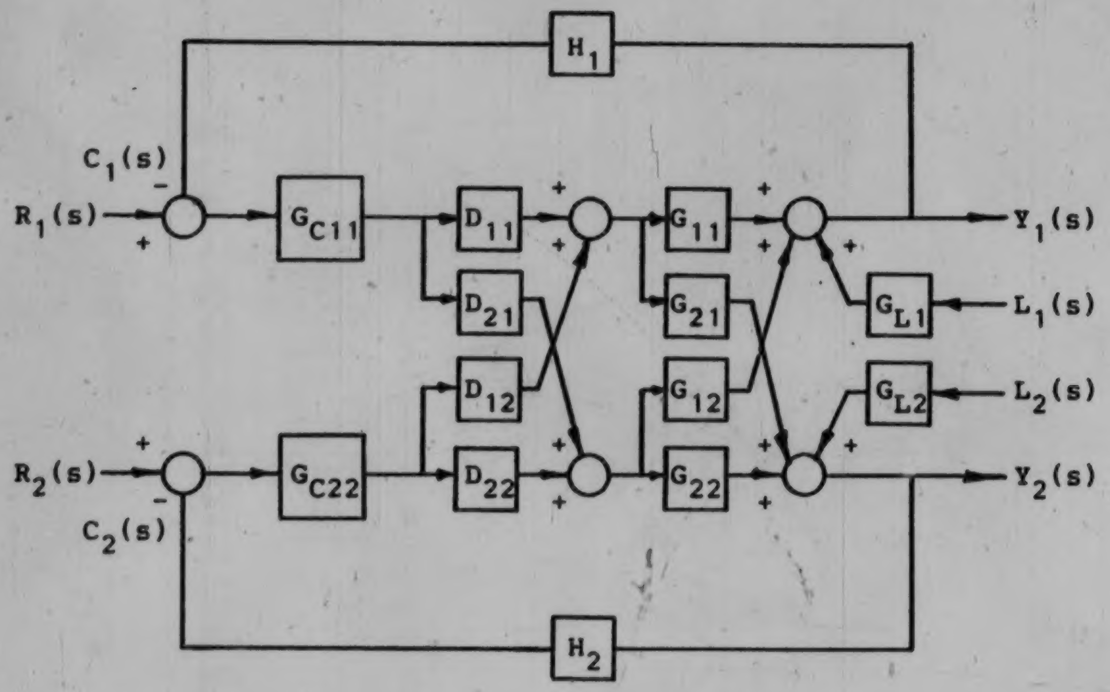

H$_1$
C$_1$(s)
R$_1$(s)
+
−
G$_{C11}$
D$_{11}$
D$_{21}$
D$_{12}$
D$_{22}$
+
+
+
+
G$_{11}$
G$_{21}$
G$_{12}$
G$_{22}$
+
+
+
+
+
+
G$_{L1}$
G$_{L2}$
Y$_1$(s)
L$_1$(s)
L$_2$(s)
Y$_2$(s)
R$_2$(s)
+
−
C$_2$(s)
G$_{C22}$
H$_2$

Figure 2.15: Block Diagram for a General Noninteracting Control Scheme

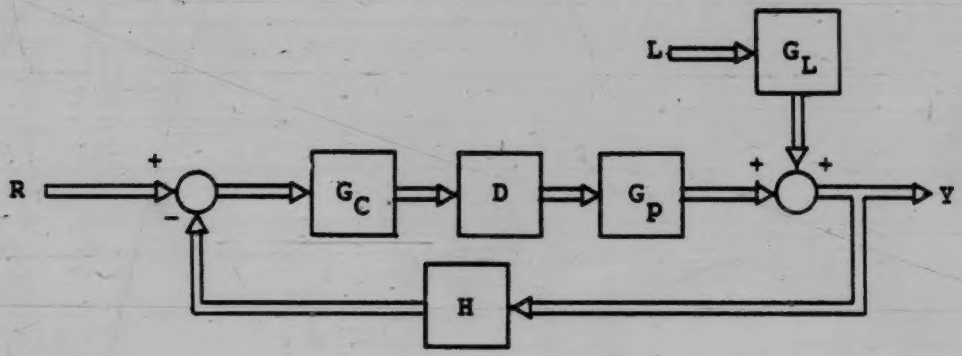


Figure 2.16: Equivalent Representation of a Block Diagram for a General Noninteracting Control Scheme

$$P_{11}=\frac{\begin{array}{l}[G_{11}D_{11}+G_{12}D_{21}+((G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21}\\ \quad +(G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22})G_{C22}H_2]G_{C11}\end{array}}{\begin{array}{l}[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21}\\ \quad +(G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2\end{array}} \qquad (2.63)$$

$$P_{12} = \frac{(G_{11}D_{12} + G_{12}D_{22})G_{C22}}{\begin{array}{l}[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21}\\ \quad +(G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2\end{array}} \qquad (2.64)$$

$$P_{21} = \frac{(G_{21}D_{11} + G_{22}D_{21})G_{C11}}{[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21} + (G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2} \tag{2.65}$$

$$P_{22} = \frac{[G_{21}D_{12}+G_{22}D_{22}+((G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21} + (G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22})G_{C11}H_1]G_{C22}}{[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21} + (G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2} \tag{2.66}$$

and:

$$Q_1 = \frac{G_{L1}+[(G_{21}D_{12}+G_{22}D_{22})G_{L1}-(G_{11}D_{12}+G_{12}D_{22})G_{L2}]G_{C22}H_2}{[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21} + (G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2} \tag{2.67}$$

$$Q_2 = \frac{G_{L2}+[(G_{11}D_{11}+G_{12}D_{21})G_{L2}-(G_{21}D_{11}+G_{22}D_{21})G_{L1}]G_{C11}H_1}{[(G_{21}G_{12}-G_{11}G_{22})D_{12}D_{21} + (G_{22}G_{11}-G_{12}G_{21})D_{11}D_{22}]G_{C11}H_1G_{C22}H_2} \tag{2.68}$$

It is clear that there has been a dramatic increase in the complexity of these transfer functions compared with the coupled case presented in equations (2.46) through (2.51). However, an appropriate choice for the decoupler matrix will set the off diagonal elements (P_{12} and P_{21}) of the closed loop transfer function matrix to zero, thus eliminating the interactions.

The simplest form of the decoupler matrix is the inverse of the steady state process gain matrix (c.f. equation (2.56)). Thus:

$$D = A^{-1} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}^{-1} \qquad (2.69)$$

or:

$$D = \begin{bmatrix} \dfrac{K_{22}}{K_{11}K_{22}-K_{12}K_{21}} & \dfrac{-K_{12}}{K_{11}K_{22}-K_{12}K_{21}} \\ \dfrac{-K_{21}}{K_{11}K_{22}-K_{12}K_{21}} & \dfrac{K_{11}}{K_{11}K_{22}-K_{12}K_{21}} \end{bmatrix} \qquad (2.70)$$

Using the appropriate elements of equation (2.70) in equations (2.63) through (2.66) and considering only steady state plant operation, it follows that the closed loop transfer function matrix becomes:

$$P = \begin{bmatrix} \frac{2}{H_1} & 0 \\ 0 & \frac{2}{H_2} \end{bmatrix} \qquad (2.71)$$

The zero elements in the off diagonal positions indicate that the steady state interactions have been eliminated. It should also be clear that the decoupler design is dependent on reliable values of the steady state process gains (i.e. depends on model uncertainty). For processes with time varying gains, the decoupler will require changes from time to time, however, partial decoupling will always result which is usually considered better than no decoupling at all. It is important to note that equation (2.71) is not valid during transients in plant operation because the process dynamics have not been accounted for. This results in a situation similar to the time varying process gain case in that the process may be partially decoupled during the transient and if a new steady state is reached, model mismatch in the decoupler could result from nonlinear gains and once again the decoupler could require changes for perfect operation.

With consideration of the aforementioned problems, it becomes evident that better decoupling behavior will result if the decoupler is designed based on a dynamic model of the plant. One of the original methods is that of Boksenbom and

Hood (1949). These authors suggest using the control system outlined in the block diagram given in Figure 2.17. As stated earlier, the objective of the decoupling controllers (G_{C12} and G_{C21} is to make the off-diagonal elements in the closed loop transfer funcion matrix identically zero. With this in mind, analysis of the block diagram in Figure 2.17 leads to:

$$G_{C12} = \frac{-G_{12}G_{C22}}{G_{11}} \tag{2.72}$$

and:

$$G_{C21} = \frac{-G_{21}G_{C11}}{G_{22}} \tag{2.73}$$

as the required decoupler forms. Details of the derivation of these controllers has been presented by Wood (1977). It is interesting to note that the closed loop transfer function for a system using this decoupling technique remains unchanged from the interacting case (c.f. equation 2.42) except for the controller transfer function matrix which becomes:

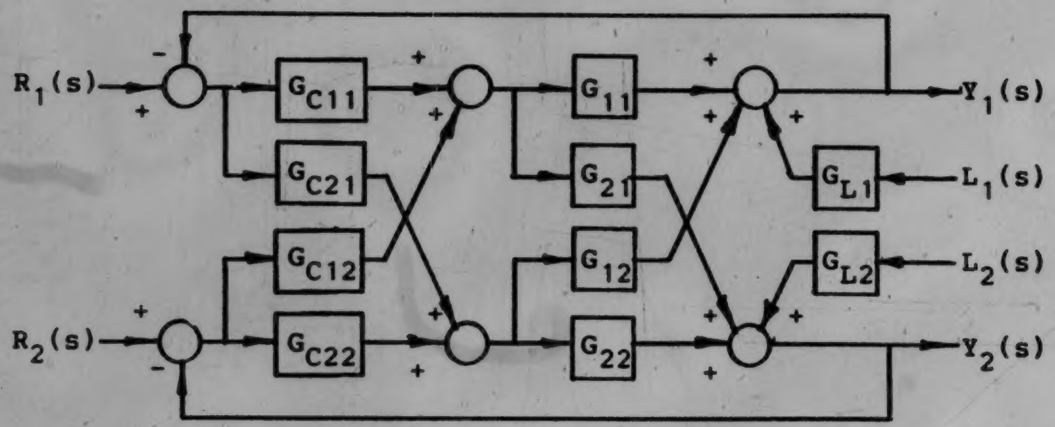

$R_1(s)$
$R_2(s)$
G_{C11}
G_{C21}
G_{C12}
G_{C22}
G_{11}
G_{21}
G_{12}
G_{22}
G_{L1}
G_{L2}
$L_1(s)$
$L_2(s)$
$Y_1(s)$
$Y_2(s)$
+
-

Figure 2.17: A Block Diagram for the Boksenbom and Hood Noninteracting Control Scheme

$$G_C = \begin{bmatrix} G_{C11} & G_{C12} \\ G_{C21} & G_{C22} \end{bmatrix}$$

To implement this decoupler, one chooses the main controllers based on SISO design criteria and estimates controller constants for them. Then, the decoupling controllers can be calculated from the above two equations and on-line tuning is performed. It should be noted that the dependence of this decoupler form on the primary loop controller constants creates an inconvenience because the decoupling controllers must be recalculated every time the primary controller constants are changed.

Luyben (1970) describes a technique he refers to as an ideal decoupler. The block diagram for this scheme is the same as that given for the static decoupler (c.f. Figure 2.15). The objective of this scheme is to make each control loop behave as if all other loops were on manual control. The closed loop equation for this block diagram, in matrix notation, was given earlier in equation (2.61). The precompensator must be designed to give a closed loop transfer function of the following form:

$$Y = \begin{bmatrix} \frac{G_{C11}G_{11}}{1+G_{C11}G_{11}} & 0 \\ 0 & \frac{G_{C22}G_{22}}{1+G_{C22}G_{22}} \end{bmatrix} R \tag{2.74}$$

It should be noted that the disturbance part of equation (2.74) has been dropped simply for convenience. The form given in this equation will result if the condition:

$$GD = \text{Diag. } G \tag{2.75}$$

or:

$$D = G^{-1}\text{Diag. } G \tag{2.76}$$

is specified so for a two by two system, the decoupling matrix is:

$$D = \begin{bmatrix} \frac{G_{11}G_{12}}{G_{11}G_{22}-G_{12}G_{21}} & \frac{-G_{12}G_{22}}{G_{11}G_{22}-G_{12}G_{21}} \\ \frac{-G_{11}G_{21}}{G_{11}G_{22}-G_{12}G_{21}} & \frac{G_{11}G_{12}}{G_{11}G_{22}-G_{12}G_{21}} \end{bmatrix} \tag{2.77}$$

In this case, the decoupler is completely independent of the primary controller.

A third decoupling scheme is credited to Zalkind (1967). The noninteracting control system this author proposes is shown in the block diagram in Figure 2.18. It is easily seen from this diagram that ~~the~~ decoupling controllers must exactly counteract the contribution of the cross terms in the plant transfer function matrix. That is:

$$G_{11}G_{C12}G_{C22} + G_{C22}G_{12} = 0 \tag{2.78}$$

and:

$$G_{22}G_{C21}G_{C11} + G_{C11}G_{21} = 0 \tag{2.79}$$

Solving for the decoupling controllers gives:

$$G^{*}_{C12} = \frac{-G_{12}}{G_{11}} \tag{2.80}$$

and:

$$G^{*}_{C21} = \frac{-G_{21}}{G_{22}} \tag{2.81}$$

The final controller matrix in this case is represented as:

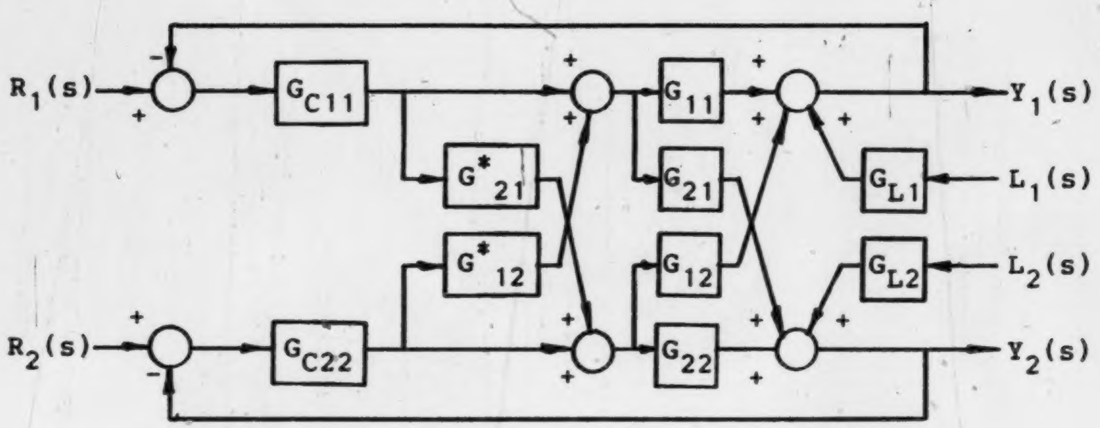

$R_1(s)$
$R_2(s)$
G_{C11}
G_{C22}
G^*_{21}
G^*_{12}
G_{11}
G_{21}
G_{12}
G_{22}
G_{L1}
G_{L2}
$L_1(s)$
$L_2(s)$
$Y_1(s)$
$Y_2(s)$
+
−

Figure 2.18: A Block Diagram for the Zalkind Noninteracting Control Scheme

$$G_C = \begin{bmatrix} G_{C11} & G_{C22}G^*_{C12} \\ G_{C11}G^*_{C21} & G_{C22} \end{bmatrix} \qquad (2.82)$$

This type of decoupler can be thought of as feedforward controllers inside the primary control loops. It is interesting to note the similarity between equation (2.82) and the corresponding controller matrix used in Boksenbom and Hood's decoupling technique. The main difference is the origin of the controller input signal. In this case, the outputs from the primary loop controllers are used as the inputs to the decoupling controllers which leads to independence from the primary controller. Consequently, changes in primary controller constants do not affect the form of the decoupling controllers and so on-line tuning causes fewer problems.

It is to be noted that the controller given by equations (2.80) and (2.81) are the same as those proposed by Luyben (1970) for his simplified decoupler. Luyben noted that this form was not outperformed by the ideal decoupler and in fact, a more robust decoupler resulted under certain conditions.

The realizability of all three of these decouplers is highly dependent on the transfer function model. Luyben's technique can easily result in a very complex decoupler that is economically prohibitive to implement. Clearly, if the process transfer function matrix model is composed of second order elements, the decoupler will be fourth order. This problem manifests itself at a much slower rate in either of the other two techniques presented in this section because individual elements of the process model are not being multiplied by each other.

Time delays in the process have the potential to cause problems with physical realizability. If any of the time delays in the diagonal elements of the process transfer function matrix model are greater than those in the off (row) diagonal elements, the decouplers of Zalkind and Boksenbom and Hood become predictive in nature and can not be physically implemented. In such a case, it is common practice to ignore the time delays that cause the problem by designing and using decoupling controllers based on this delayless model (Smith and Corripio, 1985). The impact of time delays on the physical realizability of Luyben's ideal decoupler is less clear because time delays may cancel each other. It is interesting to note that in the original work of Luyben (1970) and Zalkind (1967), the effect of process time delays on their decoupling methods was not considered.

It should be made clear that a lead/lag network cascaded with a gain element and a time delay element will be the norm for implementation of all three of these methods. The order of the overall compensator is directly related to the order of the process transfer function representation used. A noteworthy point here is that most processes can be adequately modeled by a set of first order plus time delay transfer functions. Appendix E presents a discussion on the design of a decoupler for a two by two interacting system using Zalkind's technique. Several different simulations are performed and analyzed so that an appreciation of the idiosyncracies of this type of decoupling can be gained.

2.6.4 Time Delay Compensation

Process time delays are a well known source of instability in feedback control systems. Time delay compensation arises from the fact that tighter control is possible in a dynamically equivalent system where time delays are not present. A great majority of compensation techniques involve the use of a predictive type controller that is based on a dynamic model of the process.

One of the early techniques, known as the Smith predictor, was developed for SISO control systems where a well known process model with only a single, time invariant time delay existed. The objective of this compensation method is to predict, into the future, values of the process output for a time equal to the time delay. Using this undelayed signal in the feedback loop allows the controller to immediately see how the current control action will affect the process. In effect, the feedback controller is controlling an undelayed model of the process, rather than the actual process.

The theoretical development for this technique follows from an analysis of the SISO control system block diagram in Figure 2.19. The most general case has time delays associated not only with the plant, but also with the measurement device as shown in the figure. In other words, the process and the measurement device can be represented with dynamic models having the following form:

$$G_p(s) = G^*_p(s)e^{-a_1 s} \tag{2.83}$$

and:

$$G_m(s) = G^*_m(s)e^{-a_2 s} \tag{2.84}$$

respectively. It should be noted that $G^*_p(s)$ and $G^*_m(s)$ have no internal time delays (i.e. a_1 and a_2 contain all the

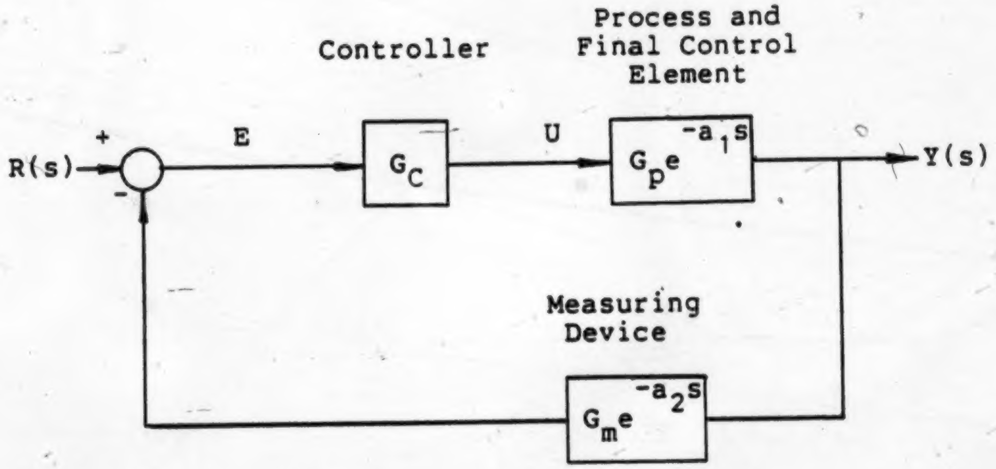

Controller
Process and Final Control Element
R(s)
+
-
E
G_C
U
$G_p e^{-a_1 s}$
Y(s)
Measuring Device
$G_m e^{-a_2 s}$

Figure 2.19: Block Diagram for a SISO Control System with Process and Measurement Time Delays

associated delays).

The usual feedback signal for this system is:

$$C(s) = G_m(s)G_p(s)G_c(s)R(s) \tag{2.85}$$

but to overcome the time delay an undelayed feedback signal to the controller, given by:

$$C^*(s) = G^*_m(s)G^*_p(s)G_c(s)R(s) \tag{2.86}$$

that contains no time delays is required. This signal can be obtained if the signal given by:

$$C_k(s)=(G^*_m(s)G^*_p(s)-G_m(s)G_p(s))G_c(s)R(s) \tag{2.87}$$

or:

$$C_k(s)=(1-e^{-(a_1+a_2)})G^*_m(s)G^*_p(s)G_c(s)R(s) \tag{2.88}$$

is added to equation (2.85). This signal uses models of the actual elements and assumes that the process model ($G^*_p(s)$ and a_1) and the measurement device model ($G^*_m(s)$ and a_2) are accurate representations of system behavior. Clearly, model error (mismatch) will degrade the effectiveness of the time delay compensation with errors in the magnitude of the time delays (a_1 and a_2) causing much more severe problems than dynamic modelling errors ($G^*_p(s)$ and $G^*_m(s)$). To overcome this problem, Vogel and Edgar (1980) have developed a more advanced technique based on the Smith predictor, that does not require that the time delay be known explicitly.

Implementation of the Smith predictor compensator is shown in the block diagram in Figure 2.20. The block labelled "time delay compensator" in the diagram incorporates the model of the process and generates the signal described by equation (2.88). As mentioned earlier,

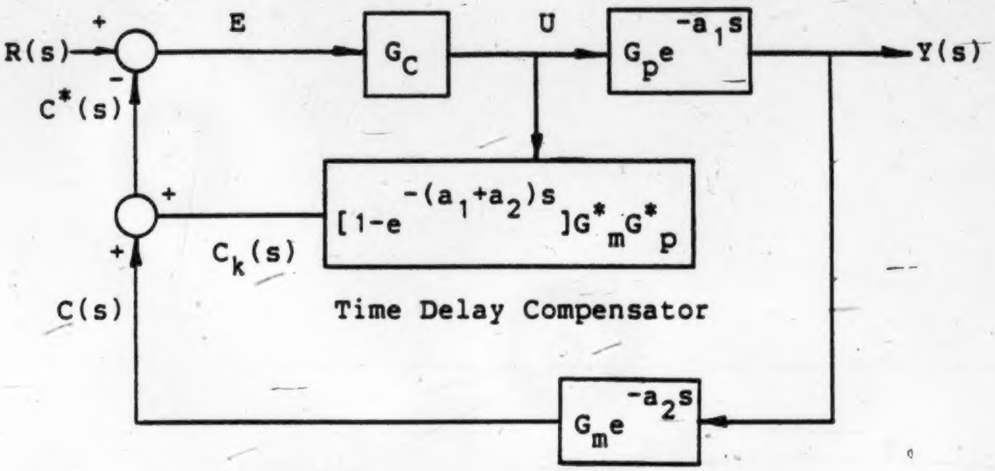


Figure 2.20: Block Diagram for a SISO Control System with a Smith Predictor Time Delay Compensator

this model should match the real process as closely as possible to avoid model mismatch.

Obviously it should be advantageous to apply some sort of time delay compensation in the control of a multivariable system. Many researchers have produced several different techniques in the last decade. Of particular note, Alevisakis and Seborg (1973, 1974) have extended the Smith predictor to the multivariable case where there is a single time delay (i.e. all elements of the process transfer function matrix have the same time delay associated with them). Ogunnaike and Ray (1979) have taken this one step further and have devised a compensation method based on the Smith predictor that is capable of handling multivariable systems with multiple time delays. This technique can be developed in a similar fashion to the original Smith predictor. A general, multivariable feedback control system including a multivariable, multidelay compensator is presented in the block diagram in Figure 2.21. The time delay compensation block shown as the block labelled G_k in the diagram can easily be derived if equations (2.85) through (2.87) are developed for a MIMO system. The usual feedback signal for this system is:

$$C(s) = G_m(s)G_p(s)G_c(s)R(s) \tag{2.89}$$

where:

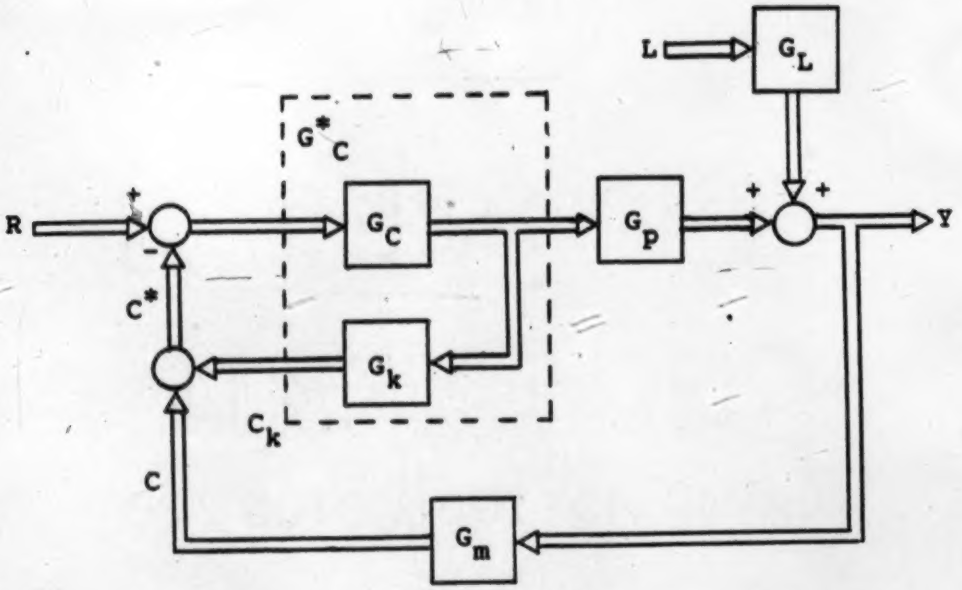


Figure 2.21: Block Diagram for a General Multivariable Multidelay Compensator Feedback Control System

$$C(s)=\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

$$G_m=\begin{bmatrix} G_{m1}e^{-\alpha_{m1}s} & 0 \\ 0 & G_{m2}e^{-\alpha_{m2}s} \end{bmatrix}$$

$$G_p=\begin{bmatrix} G_{11}e^{-\alpha_{11}s} & G_{12}e^{-\alpha_{12}s} \\ G_{21}e^{-\alpha_{21}s} & G_{22}e^{-\alpha_{22}s} \end{bmatrix}$$

$$G_c=\begin{bmatrix} G_{C11} & 0 \\ 0 & G_{C22} \end{bmatrix}$$

$$R=\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

The required undelayed signal is:

$$C^*(s)=G^*_m(s)G^*_p(s)G_c(s)R(s) \qquad (2.90)$$

where:

$$G^*_m = \begin{vmatrix} G_{m1} & 0 \\ 0 & G_{m2} \end{vmatrix}$$

$$G^*_p = \begin{vmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{vmatrix}$$

and can be generated by the addition to equation (2.89) of:

$$C_k(s)=(G^*_m(s)G^*_p(s)-G_m(s)G_p(s))G_c(s)R(s) \qquad (2.91)$$

where:

$$C_k(s)= \begin{vmatrix} C_{k1} \\ C_{k2} \end{vmatrix}$$

so $G_k(s)$ is given by:

$$G_k = G_m{}^*G^*{}_p - G_mG_p \qquad (2.92)$$

Implementation of this compensator for a two by two interacting system is shown in Figure 2.22.

The ramifications of this compensation technique can be demonstrated by developing the characteristic equation for the compensated closed loop transfer function. The inner loop, shown in Figure 2.22 as the dotted box labelled G^*_C, is first analyzed as follows:

$$\frac{U}{E} = G^*_C \qquad (2.93)$$

and:

$$G^*_C = [I + G_CG_k]^{-1}G_C \qquad (2.94)$$

Now, writing the transfer function for the entire system as:

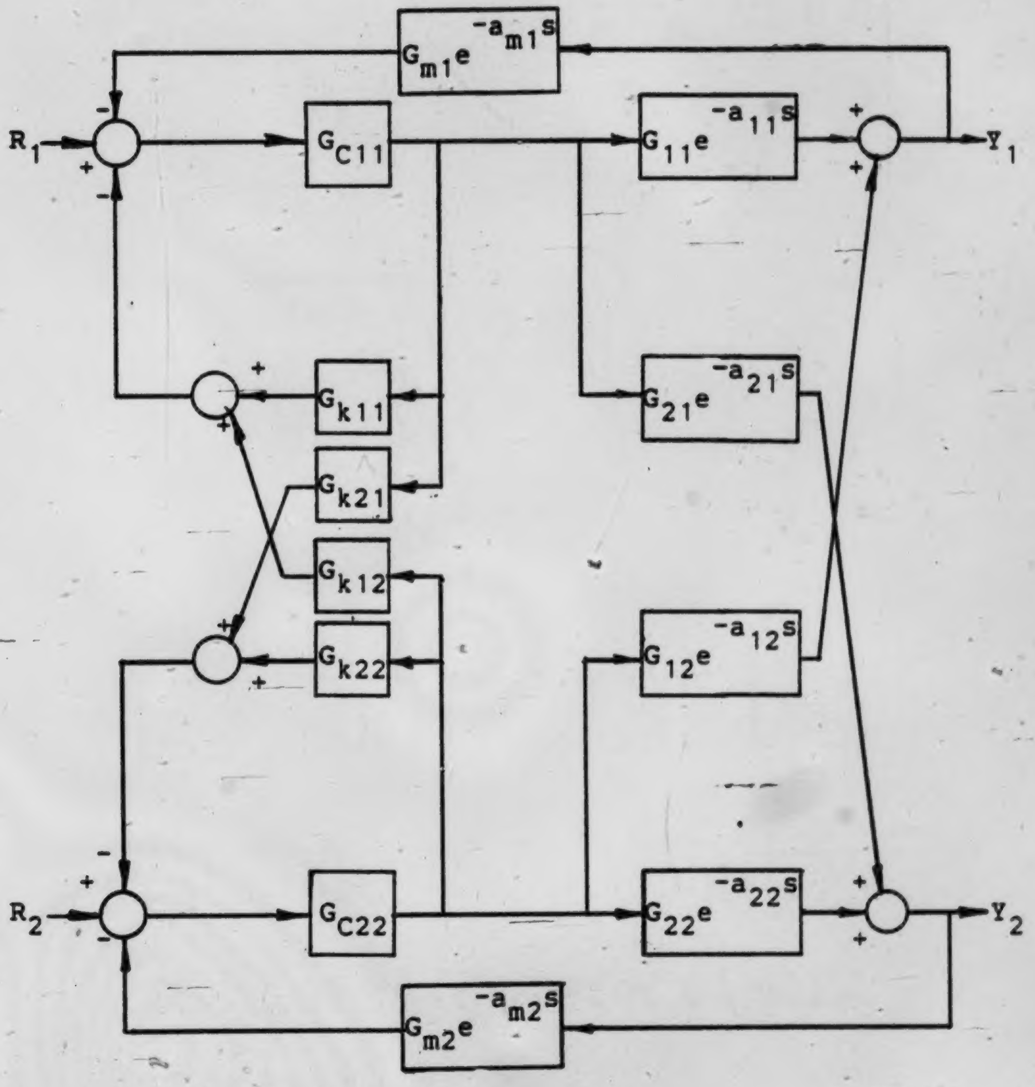

$G_{m1}e^{-a_{m1}s}$
R_1
Y_1
G_{C11}
$G_{11}e^{-a_{11}s}$
G_{k11}
$G_{21}e^{-a_{21}s}$
G_{k21}
G_{k12}
G_{k22}
$G_{12}e^{-a_{12}s}$
R_2
Y_2
G_{C22}
$G_{22}e^{-a_{22}s}$
$G_{m2}e^{-a_{m2}s}$
+
-

Figure 2.22: Block Diagram for a Two by Two Time Delay Compensated Control System

$$Y = [I+G_pG^*_CG_m]^{-1}[G_pG^*_CR+G_LL] \tag{2.95}$$

and then substituting for G_k from equation (2.92) and G^*_C from equation (2.94) gives:

$$Y=[I+G_pM^{-1}G_CG_m]^{-1}[G_pM^{-1}G_CR+G_LL] \tag{2.96}$$

where:

$$M = I + G_C(G^*_mG^*_p - G_mG_p) \tag{2.97}$$

or:

$$M+G_CG_mG_p = I+G_CG^*_mG^*_p \tag{2.98}$$

If G_p is square and nonsingular, the following identity:

$$[I+G_pM^{-1}G_CG_m]^{-1}=G_p[M+G_CG_mG_p]^{-1}RG_p^{-1} \tag{2.99}$$

can be used in equation (2.96) along with equation (2.98) to give the overall closed loop transfer function:

$$Y=G_p[I+G_CG_m^*G^*_p]^{-1}G_CR+G_p[I+G_CG_m^*G^*_p]^{-1}MG_p^{-1}G_LL \tag{2.100}$$

The stability of the closed loop system, determined by the characteristic equation:

$$|I+G_CG_m^*G^*_p| = 0 \tag{2.101}$$

has no time delays so the compensated closed loop system will be more stable than the uncompensated system. An example using this compensation technique on an idealized two by two transfer function model is given in Appendix F.

2.7 Adaptive Control

2.7.1 Multivariable Self-Tuning Control Law

The self-tuning control law used in this work is derived from a controlled autoregressive moving average (CARMA) process model of the following form:

$$A(z^{-1})Y(t)=z^{-k_{ij}}B(z^{-1})U(t)+C(z^{-1})\Xi(t)$$
$$+z^{-d_{ij}}D(z^{-1})V(t) \qquad (2.102)$$

where:

$A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$, and $D(z^{-1})$ are m x m polynomial matrices in the backshift operator, z^{-1},

$Y(t)$ is an m x 1 vector of measurable process outputs,

$U(t)$ is an m x 1 vector of controller inputs to the process,

$V(t)$ is an m x 1 vector of measurable disturbances,

$\Xi(t)$ is an m x 1 vector of random noise,

k_{ij} is a time delay between the i^{th} input and the j^{th} output expressed in multiples of the sample interval,

and:

d_{ij} is a time delay between the i^{th} measurable disturbance and the j^{th} output expressed in multiples of the sample interval.

Langman (1987) states that the objective of the control law is to minimize the cost functional:

$$J=E\{[PY(t+k_{ii})-RW(t)]^T[PY(t+k_{ii})-RW(t)]$$
$$-[Q'U(t)]^T[Q'U(t)]\}. \qquad (2.103)$$

with respect to the control effort, U(t). It should be noted that P, Q', and R in equation (2.103) are weighting transfer function matrices in the backshift operator, z^{-1}. By manipulating equation (2.102) Langman (1987) derives a technique for estimating $Y(t+k_{ii})$, the process output k_{ii} steps into the future, which allows him to minimize the cost functional to yield the control law:

$$\begin{aligned}[z^{(k_{ii}-k_{ij})}\hat{G}+Q]U(t) = {} & RW(t)-P_d\hat{F}Y(t)- \\ & z^{(k_{ij}-d_{ij})}\hat{L}V(t)- \\ & \hat{H}PY^*(t+k_{ii}-1|t-1) \end{aligned} \tag{2.104}$$

where $\hat{G}$, $\hat{F}$, $\hat{H}$, and $\hat{L}$ are the estimated polynomial matrices associated with A, B, C, and D, respectively. The term $Y^*(t+k_{ii}|t)$ is the predicted output of the process k_{ii} steps ahead into the future. A detailed account of the derivation of the control law in equation (2.104) is given by Langman (1987), and will not be repeated here.

2.7.2 Parameter Estimation

The key to the adaptive control law presented in the last section is the parameter estimation scheme used to determine the process model parameters. Langman (1987) suggests that the recursive least squares (RLS) method with upper diagonal factorization (UD) of the covariance matrix will produce good results. For a summary of the RLS algorithm and additional details on parameter estimation see Langman (1987).

3. Grinding Circuit Dynamic Model

3.1 Introduction

A conventional rod mill - ball mill wet grinding circuit, in general, consists of a combination of the following items of process equipment:

(i). Grinding Mills
(ii). Hydrocyclones
(iii). Pump and Sump
(iv). Conveyors

In order to apply the advantages afforded by digital simulation to study such circuits, mathematical models of each of the above unit operations are needed.

Over the past two decades, much work has been done to develop the required models. The purpose of this chapter is to describe the models used for studying the dynamics and control of grinding circuits in work at the University of Alberta initiated by Wood and Flintoff (1982). The models used in the simulator for the major unit operations follow those outlined by Smith and Guerin (1980). There are two reasons for this choice. First, the models are relatively simple which eases both understanding and computational effort, and second, the model parameters for an operating circuit, that of the Lake Dufault division of Falconbridge Copper Mines Limited, are provided by the authors. It should be noted here that these types of models have been applied to other grinding circuits by other workers, for example Lynch (1977), Rajamani and Herbst (1980), and Bascur, Freeh and Herbst (19[illegible]5).

The next section of this chapter gives some background on the concepts involved in the size reduction of solid materials and describes a differential equation model that can be used to simulate a grinding mill. Section 3.3 contains a brief description of a typical hydrocyclone classifier and presents the system of nonlinear algebraic equations that constitute the model. A static pump model, which requires the solution to the hydrocyclone pressure flow relation, is discussed in Section 3.4 along with a model capable of simulating sump behavior. Fixed and variable delay models, associated with conveyors and pipelines respectively, are treated in Section 3.5. Finally, Section 3.6 deals with various methods of modelling changes in ore hardness.

3.2 Grinding Mills

3.2.1 Grinding Machinery

Rod and ball mills, shown schematically in Figures 3.1 and 3.2, are employed in various combinations in wet grinding operations. The coarse feed slurry, consisting of water and solids, enters both types of mills through a scoop box or chute. From there, the slurry is gravity fed into the rotating mill where it is ground with either steel rods or balls.

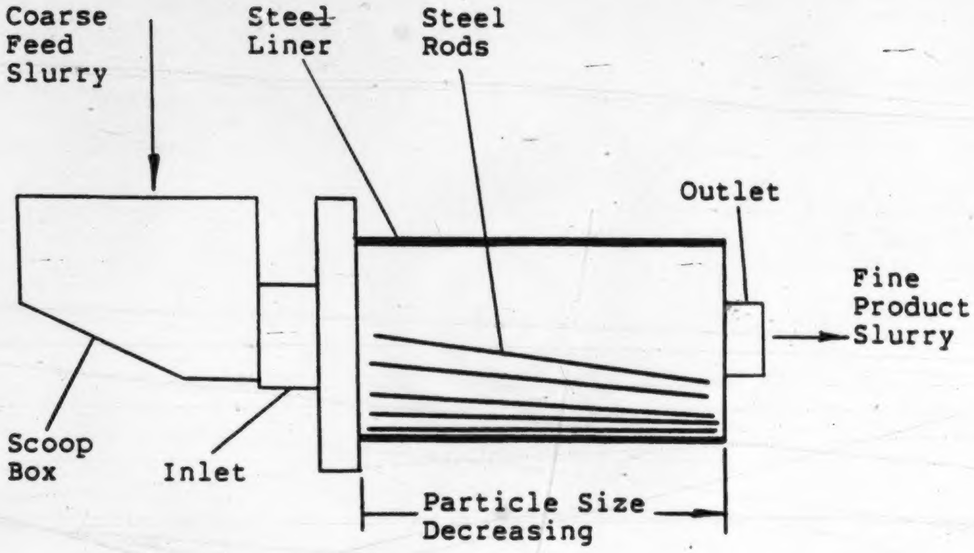

Coarse Feed Slurry
Steel Liner
Steel Rods
Outlet
Fine Product Slurry
Scoop Box
Inlet
Particle Size Decreasing

Figure 3.1: Schematic Diagram of a Typical Rod Mill

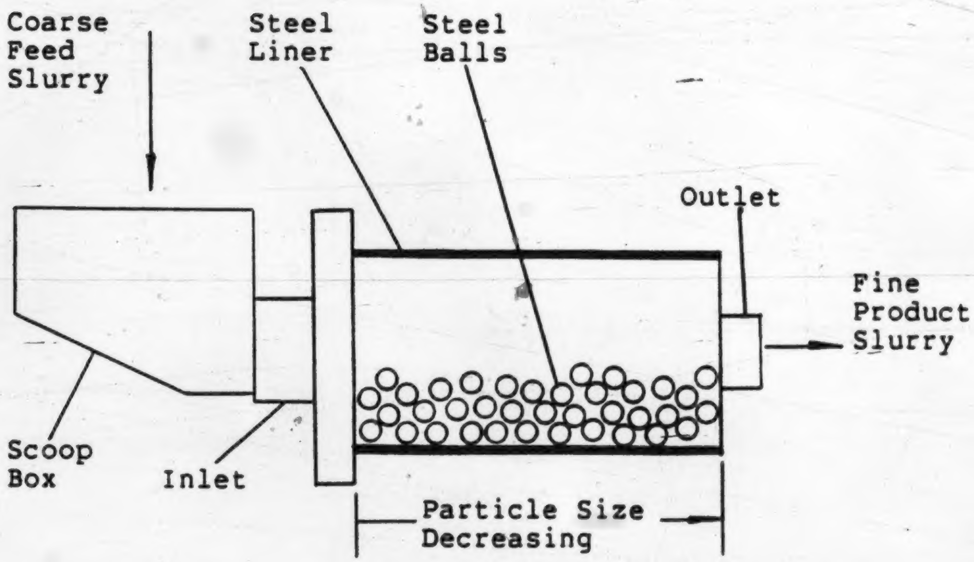

Coarse Feed Slurry
Steel Liner
Steel Balls
Outlet
Fine Product Slurry
Scoop Box
Inlet
Particle Size Decreasing

Figure 3.2: Schematic Diagram of a Typical Ball Mill

Rod mills are used to grind coarse feed ranging from 80% passing 20 mm to 80% passing 4 mm into a product size ranging from 80% passing 2 mm to 80% passing 0.5 mm. The physical size of this type of mill is limited by the 6.8 metre practical length of rods due to flexing and breakage problems (Rowland, 1982). The length to diameter ratio of a typical rod mill is greater than 1.33:1 so that rod tangling can be avoided. The rods wear down to an eliptical cross section and will break up and pass through the mill discharge when the major axis diameter becomes about 2.5 cm (Lynch, 1977). Steel or rubber liners are used to protect the mill shell and must be replaced periodically.

Ball mills are used to produce product particle sizes ranging from 80% passing 0.5 mm to finer than 80% passing 75 micrometres. Ball mills ordinarily are similar in size to rod mills. However, by nature of the grinding media employed in this type of mill (i.e. steel balls), the physical dimensions are free of the rod mill constraints. The length to diameter ratio typically ranges from less than 1:1 to more than 2:1, depending on the particular application (Rowland, 1982). Energy use considerations usually limit the mill diameter to 5 to 6 metres. The diameter of the steel balls will depend on the application (expected feed size, desired product size, and ore hardness), with a range from 127 mm (5 inches) to a worn ball size of 16 mm (5/8 inches) (Rowland, 1982). As with rod mills, steel or rubber liners are used to protect the mill shell from damage.

3.2.2 Comminution

Several different functions can be served through comminution (particle size reduction) in the mineral processing industry. The production of product with a large surface area available for chemical reactions to occur or to free valuable minerals from waste material before concentration is of primary importance. The objective in both cases is the production of particles that consistently fall into a particular size class specification.

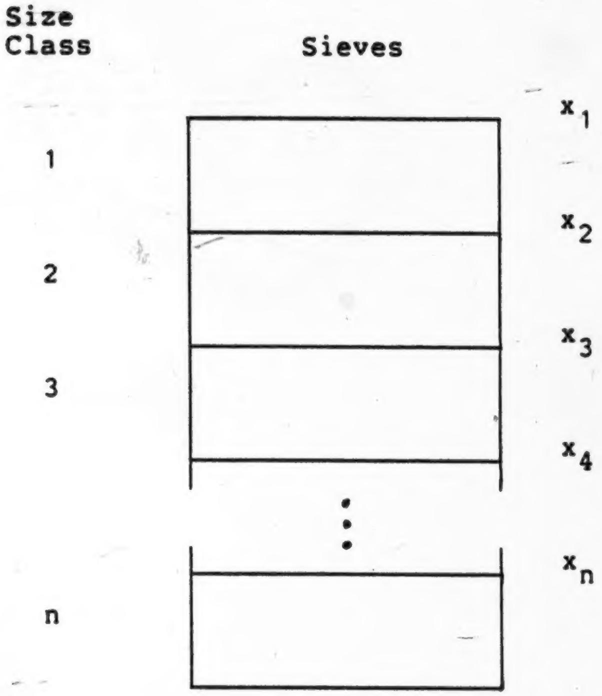
Particle size class specifications naturally arise from techniques used for particle size measurement and the subsequent mathematical representation of the size distribution. Typically, particles are measured in terms of the minimum square aperture through which it will pass. This measure is known as the sieve diameter of the particle.

There are several methods available for mathematically describing the particle size distribution. The most obvious is to postulate a continuous function and fit the parameters to sieve analysis data. A commonly used function is the Gaudin-Schuhmann distribution expressed as follows:

$$B = (x/k)^{\alpha} \; ; \; x < k \qquad (3.1)$$

where B is the cumulative weight fraction passing size x and k is the size modulus at which B is defined to be unity. The parameter, α, is known as the distribution modulus. Since the function is, by definition, continuous over the interval $0 \leq x \leq k$, the number of size classes will approach infinity which leads to an excessive amount of complexity. By fitting this function to a finite number of size data points, approximations are introduced and large errors can result. This is often unacceptable. A more appropriate form for representing a particle size distribution is to use a finite number of particle size classes. It is convenient to define the size class boundaries by using a series of consecutively smaller sieve screen sizes as points of reference as shown by the schematic diagram in Figure 3.3. The mass fraction of the particles that are retained by a given screen corresponds directly to the lower bound of a size class that "passes" the previous screen in the series. This form of discrete representation is capable of describing all possible size distributions and does not necessitate the use of approximations to fit sieve analysis data to a continuous function.

Comminution processes have historically been studied in terms of grinding mill energy consumption (Lynch, 1977) leading to empirical steady state design models. This, however, has proved to be a severely limited view of grinding in that only a small proportion of the original energy input to the process is used in actually breaking up the particles. Energy is dissipated in other forms such as sound, friction (heat), and elastic and plastic deformation


Size Class
Sieves
1
2
3
n
x_1
x_2
x_3
x_4
x_n

x_1 - size which all material just passes

x_n - passing size of finest screen

Figure 3.3: Sieve Tray Analysis for Discrete Particle Size Distribution Reference Points

of the particles. In the case of tumbling mills, which are the most common form of mill used in comminution, potential and kinetic energy losses are also experienced. Little is known about these latter two factors because of the difficulties associated with performing an internal energy balance on a tumbling type mill. This clearly indicates the deficiency in considering only energy input when attempting to derive dynamic models for use in simulation work.

Researchers continue to analyze the phenomenon of particle breakage in an attempt to relate the breakage process mathematically to mill operating parameters such as feed particle size distribution, feed slurry flow rate and mass fraction of solids, and particle hardness. The difficulty in adequately describing breakage in an industrial operation of a size reduction machine such as a tumbling mill stems from the fact that particles do not undergo simple primary breakage alone (i.e. only a single breakage event) but a succession of breakage events. The formulation used to describe the overall breakage operation mathematically arises from defining the rates at which particles of various size classes undergo breakage, the size distribution, by mass, expected from the breakage of a unit of mass of one particle size class into smaller classes, and the repetitive nature of the process. It should be noted that this modelling approach known as "phenomenological modelling".

3.2.3 Grinding Mill Model

It has been found that the same general form of model can be used to approximately describe the particle reduction dynamics in both rod and ball mills (Smith and Guerin, 1980). The model is based on a perfect mixer transport model with first order breakage kinetics. An unsteady state particle size class frequency balance around a grinding mill can be written in a similar fashion as for a chemical species concentration balance on a continuous stirred tank reactor. A general unsteady state material balance:

$$\text{Input} + \text{Generation} = \text{Output} + \text{Accumulation} \tag{3.2}$$

results in the basic model form. The input and output terms in equation (3.2), for a particular particle size class "i", are simply:

$$\text{Input} = F_s f_i \tag{3.3}$$

and:

$$\text{Output} = P_s p_i \tag{3.4}$$

The rate of generation of particles of this size is determined from the rate at which these particles are broken into smaller particles and the rate at which particles in the i-1 larger size classes are broken down into the ith size class. In other words:

$$\text{Generation} = -k_i H_s p_i + \sum_{j=1}^{i-1} b_{ij} k_i H_s p_i \tag{3.5}$$

It should be noted that this equation reflects the perfect mixer transport assumption, where the mill product and internal compositions are identical. The accumulation term represents the rate of change of particles in size class "i" that are present in the mill given:

$$\text{Accumulation} = \frac{d(H_s p_i)}{dt} \tag{3.6}$$

Expanding equation (3.6) yields:

$$\frac{d(H_s p_i)}{dt} = p_i \frac{dH_s}{dt} + H_s \frac{dp_i}{dt} \tag{3.7}$$

The mass holdup of solids in the mill is:

$$H_s = Vc_s \tag{3.8}$$

from which it follows that:

$$\frac{dH_s}{dt} = V\frac{dc_s}{dt} + c_s\frac{dV}{dt} \tag{3.9}$$

It is standard practice, in the case of high volumetric throughput mills, to assume a constant volumetric holdup which leads to:

$$\frac{dV}{dt} = 0 \tag{3.10}$$

and implies that the inlet and outlet volumetric flow rates are equal. As well, for the purposes of this work, changes in concentration of solid material in the mill will be assumed negligible, so that:

$$\frac{dc_s}{dt} = 0 \tag{3.11}$$

Combining equations (3.10) and (3.11) with (3.9) and (3.7), it follows that:

$$\text{Accumulation} = H_s \frac{dp_i}{dt} \tag{3.12}$$

Using equations (3.3) through (3.5) and equation (3.12) in equation (3.2) gives:

$$F_s f_i - k_i H_s p_i + \sum_{j=1}^{i-1} b_{ij} k_j H_s p_j = P_s p_i + H_s \frac{dp_i}{dt} \tag{3.13}$$

The model that describes the rate of change of particles of size x_i in the mill follows from equation (3.13):

$$\frac{dp_i}{dt} = \frac{F_s f_i - P_s p_i}{H_s} - k_i p_i + \sum_{j=1}^{i-1} b_{ij} k_i p_i \tag{3.14}$$

The final form of the model results from the realization that the inlet and outlet mass flow rates are equal, due to the constant holdup assumption, and are related to the volumetric flow rate by:

$$F_s = P_s = Qc_s \tag{3.15}$$

Then, using equations (3.8) for the mass holdup, equation

(3.14) can be written as:

$$\frac{dp_i}{dt} = \frac{f_i - p_i}{\tau} - k_i p_i + \sum_{j=1}^{i-1} b_{ij} k_i p_i \qquad (3.16)$$

where the mean residence time of the mill is:

$$\tau = V/Q \qquad (3.17)$$

with the mill volumetric holdup considered as the effective mill volume, or pulp volume in the mill after the exclusion of the rod or ball loading. This volume is normally estimated by impulse tracer testing on the liquid phase.

Expanding the mill model, equation (3.16), for the entire population of size classes results in the complete model:

$$\frac{dp_i}{dt} = \frac{f_i - p_i}{\tau} - k_i p_i + \sum_{j=1}^{i-1} b_{ij} k_i p_i \qquad (3.18)$$

for:

$i = 1, \ldots, n-1$

and:

$$p_n = 100 - \sum_{i=1}^{n-1} p_i \qquad (3.19)$$

It should be noted that the particle size classification number (subscript i) in the above equations is an index that references a specific minimum particle size as outlined in Section 3.2.2.

The normalized breakage function, b_{ij}, and the selection function, k_i, are the parameters that play the key role in distinguishing between the two mill types. Neither of these parameters can be uniquely determined as the values are dependent on both the ore type and the individual mill (Lynch, 1977). One approach to overcoming this problem is to assume that the breakage function is dependent only on the particular ore, while the selection function accounts for the machine characteristics.

The breakage function used in this work follows the form of the Gaudin-Schuhmann distribution given in equation (3.1) and determines the particle size distribution of a breaking particle in each size class. The general normalized functional form describing the breakage of a particle from a larger size class into a range of smaller size classifications is given by:

$$B_i = (x_i/x_{j+1})^a \ ; \ i = j+1, \ldots, n; \ a>0 \qquad (3.20)$$

where B is the cumulative fraction of particles passing size x_i. Particular note should be made here that by definition, $B_i=1$ for $i \le j$. This implies that particle agglomeration does not occur and that no broken material remains in the original size classification. By choosing the ratio between consecutive size classes in an orderly fashion, equation (3.20) can be written as:

$$B_i = \left[\frac{1}{r}\right]^{(i-j-1)a};\ i=j+1, \ldots n;\ a>0 \qquad (3.21)$$

where r is the size class ratio given by:

$$r = x_{j+1}/x_j \qquad (3.22)$$

A geometric size progression where $r=2^{-0.5}$ is usually convenient to use because it results from standard Tyler sieve screen sizes. The discretized breakage function follows from:

$$b_{ij} = B_i - B_{i+1} \qquad (3.23)$$

which leads to the final form:

$$b_{ij} = \begin{cases} 0 & ;\ i \le j \\ [2^{-0.5}]^{(i-j-1)a} - [2^{-0.5}]^{(i-j)a} & ;\ i>j \end{cases} \qquad (3.24)$$

$$(i,j = 1, \ldots, n)$$

which is a lower triangular matrix with zeros on the diagonal. It has been found that 14 to 16 size classifications (including the undersize or pan size) gives an adequate representation of this function (Smith and Guerin, 1980).

The parameter, a, in equation (3.24), is assumed to be dependent only on the type of ore and not the particular installation. Values for this parameter based on operating data for the Lake Dufault circuit were found to be:

$a = 0.676$; rod mill

$a = 0.652$; ball mill

The selection function is somewhat like a rate function for a chemical reaction. It determines the rate at which particles are broken from a given size class into smaller size classes and is thus dependent on the specific ore and individual grinding mill characteristics. As stated earlier, this means that in order to accurately model a particular circuit, the selection function must be fit to actual operating data. Smith and Guerin (1980) established the following functions for the Lake Dufault circuit:

Rod Mill

$$k_i = \begin{cases} 0.0212 & ;\ x_{i+1} \leq 1000\mu \\ 0.0212(x_{i+1}/1000)^{4.48} & ;\ x_{i+1} > 1000\mu \end{cases} \qquad (3.25)$$

$$(i = 1, \ldots, n-1)$$

Ball Mill

$$k_i = 1.045(x_{i+1}/1000)^{1.818} \qquad (3.26)$$

$$(i = 1, \ldots, n-1)$$

It should be noted here that a selection function for the fraction of the particles in the pan, i.e. $i=n$, is meaningless.

3.3 Hydrocyclone Classifiers

3.3.1 Classification

Separation of ground particles into different size classes can be done by two methods. The first involves screening in which particles are separated based on their size and shape. A second more common form of particle size separation is known generally as hydraulic classification. This method utilizes the movement of particles through fluids to separate them based not only on size and shape, but also on density. Hydraulic classifiers fall into two group types: mechanical and centrifugal. Rake and spiral settling tank classifiers are typical examples of the mechanical type, while the most common centrifugal form is the hydrocyclone. The advantages of hydrocyclones over their mechanical counterparts include faster, more reliable operation, and more compact size. Hydrocyclones are now the most commonly used form of classification employed in grinding operations for the aforementioned reasons (Lynch, 1977).

The nature of the operation of a hydrocyclone can be described by making reference to the schematic diagram shown in Figure 3.4. The feed stream enters the feed chamber tangentially under pressure through the inlet pipe. The cylindrical shape of the feed chamber causes the slurry to rotate, which creates centrifugal forces on the particles in the slurry. The slurry then moves down the cylindrical

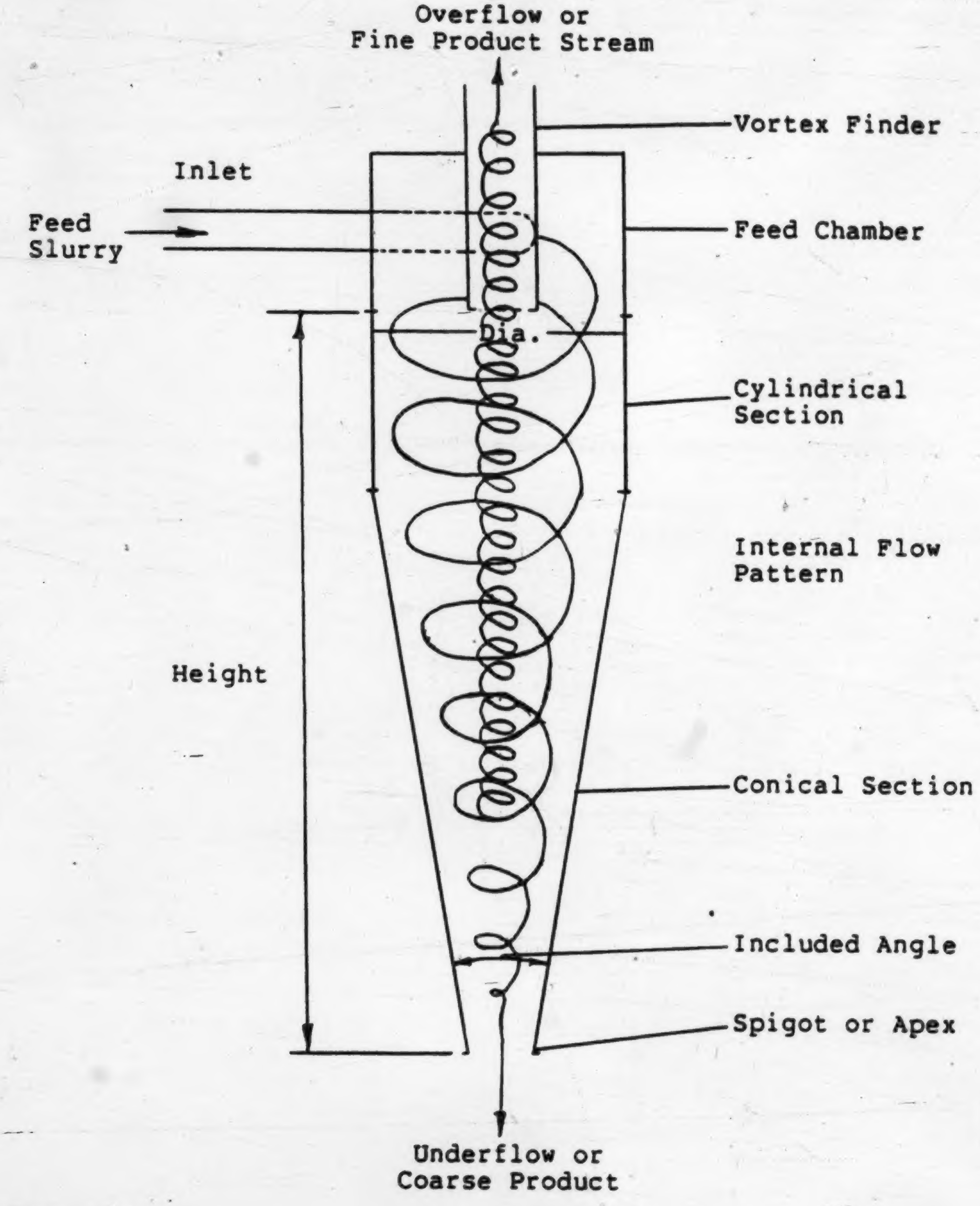

Overflow or
Fine Product Stream
Vortex Finder
Inlet
Feed
Slurry
Feed Chamber
Dia.
Cylindrical
Section
Internal Flow
Pattern
Height
Conical Section
Included Angle
Spigot or Apex
Underflow or
Coarse Product

Figure 3.4: Schematic Diagram of a Typical Hydrocyclone

section into the conical section in a spiral pattern as shown in the diagram. As the slurry proceeds down into the cone towards the apex, the smaller particles begin to migrate upwards towards the centre and spiral upwards to the vortex finder to exit as a fine product stream. The larger particles remain in the downward moving spiral path and exit as a coarse product stream through the spigot. It should be noted that both product streams, underflow and overflow, exit at or near atmospheric pressure.

Also shown on the schematic diagram in Figure 3.4 are many of the important design variables such as the inlet pipe diameter, the vortex finder diameter, the spigot (or apex) diameter, and the feed chamber diameter. The combined height of the cylindrical and conical sections below the vortex finder determines the hydrocyclone retention time. All of these variables affect either the efficiency or capacity of a given unit. The reader is refered to Lynch (1977) and Arterburn (1982) for further details on the design specifications for hydrocyclones.

It should be clear from consideration of the diagram that the potential for short circuiting of the feed slurry to both product streams exists. As long as the vortex finder is extended far enough below the inlet opening, short circuiting to the fine product stream will not be a problem. However, there is always a small amount of the feed that bypasses classification and reports to the underflow stream.

3.3.2 Product Size

The product size of a hydrocyclone is normally specified in terms of a corrected cut size. The corrected cut size is defined as the equiprobable particle partition size or the size for which fifty mass percent of the particles in the feed slurry reports to both the overflow and underflow streams considering only centrifugal classification (i.e. classification ignoring particle bypass to the underflow stream). Figure 3.5 shows a typical relation between particle diameter and the fraction of particles in the feed stream that report to the underflow. The curve labled "actual" in this diagram

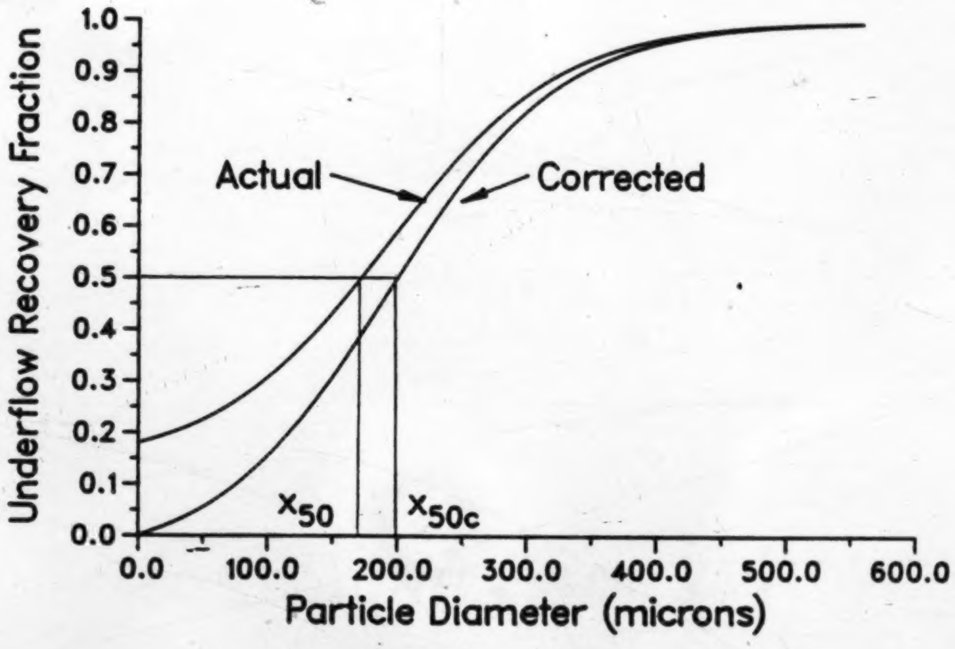


Figure 3.5: Typical Relationship Between Particle Size and Hydrocyclone Underflow Recovery

demonstrates the effect of classification bypassing that exists in every hydrocyclone. It is assumed that at least a small amount of all particles present in the feed stream are recovered in the underflow stream, thus bypassing classification. The "corrected" curve is determined from the fraction of feed fluid that reports to the underflow stream of the hydrocyclone. It has been found that the minimum fraction of solids of each size class reporting to the underflow stream corresponds directly to the fraction of the feed liquid reporting to the underflow (Lynch, 1977). Each particle size recovery fraction is corrected for bypass to produce the corrected curve. In other words, the following equation:

$$Y_{ci} = \frac{Y_{ai} - Rf}{1 - Rf} \qquad (3.27)$$

can be used to give the corrected efficiency curve where Y_{ci} indicates the corrected value of the fraction of particles of size class i that report to the underflow stream.

Because the x_{50c} value changes depending on the feed stream particle size distribution and inlet flow rate, the efficiency curves in Figure 3.5 will shift horizontally. Thus, efficiency curves for hydrocyclones are usually given in a reduced form where the particle size on the corrected curve has been divided by the x_{50c} value. A representative curve of this type of efficiency curve is shown in Figure 3.6. Efficiency curves of this form have

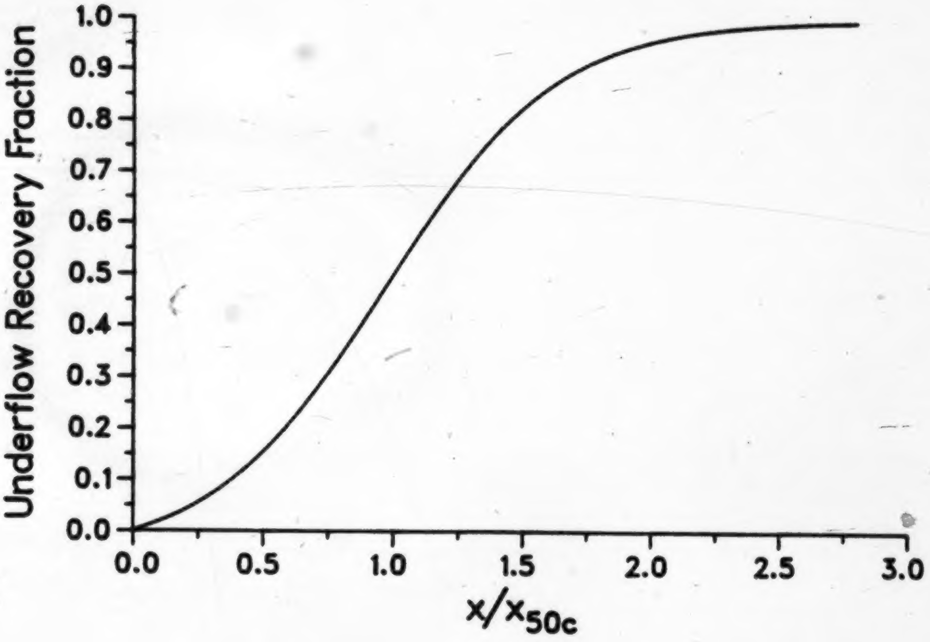


Figure 3.6: Typical Reduced Efficiency Curve for a Hydrocyclone

been found to remain constant over a wide range of operating conditions (Arterburn, 1982). It should be noted that the exact shape of the reduced efficiency curve varies from ore to ore, and the curve presented in Figure 3.6 is used only for illustrative purposes.

3.3.3 Hydrocyclone Models

Hydrocyclones can be thought of mathematically as defining the probability that a particle of a given size will appear either in the overflow or underflow streams. As the dynamics of hydrocyclones are generally considered to be rapid with respect to most other mineral processing equipment, dynamic behavior is neglected and a model consisting of a set of algebraic equations can be used.

The model used in this work is similar to the empirical model outlined by Lynch (1977) that describes the pressure-flow relationship, the water split (between the overflow and underflow streams), reduced efficiency curve, and the equiprobable partition size as a set of nonlinear equations. Although the Plitt (1976) model has some advantages in terms of model reliability, it was decided to use the Lynch model to enable comparison to Smith and Guerin's (1980) work.

The pressure-flow relation for this model is a function of the various physical dimensions of the hydrocyclone; however, the simple equation:

$$Q = K_1 P^{0.5} Ff_w^{0.125} \tag{3.28}$$

provides an adequate fit given the constant geometry in these cyclones. Although the constant, K_1, is a function of the vortex finder diameter and must be determined for any given installation from actual operating data, it has been found to remain constant over a wide range of flow rates, pressures, and feed slurry water mass fractions (Lynch, 1977).

The next step in developing the complete model is to determine the mass of water that reports to the fine product (overflow) stream versus the mass of water in the inlet feed stream. This is known as the water split and is modelled by the equation:

$$W_o = 1.1W_f + K_2 \qquad (3.29)$$

The constant, K_2, in this equation is a function of the spigot diameter and is unique to a given hydrocyclone installation. However, as with the pressure-flow relationship in equation (3.28), equation (3.29) provides reasonable results over a wide range of operating conditions.

The particle size frequency distribution in the underflow product stream is usually expressed in terms of a reduced efficiency curve for the hydrocyclone. The general shape of such a curve, corrected for centrifugal classification only, was presented earlier in Figure 3.6. Mathematically, the curve in this figure can be modelled as follows:

$$Y_{ci} = \frac{e^{\gamma y}-1.0}{e^{\gamma y}+e^{\gamma}-2} \qquad (3.30)$$

where:

$$y = x_{i+1}/x_{50c} \qquad (3.31)$$

It should be noted that this curve is independent of the physical dimensions of the hydrocyclone and its operating conditions, and runs through the origin. The variable parameter, γ, is a measure of the sharpness of separation for a particular installation. Figure 3.7 indicates how the shape of the curve changes depending on the value of γ.

The equiprobable particle partition size, x_{50c}, is a function of the dimensions and operating conditions of the

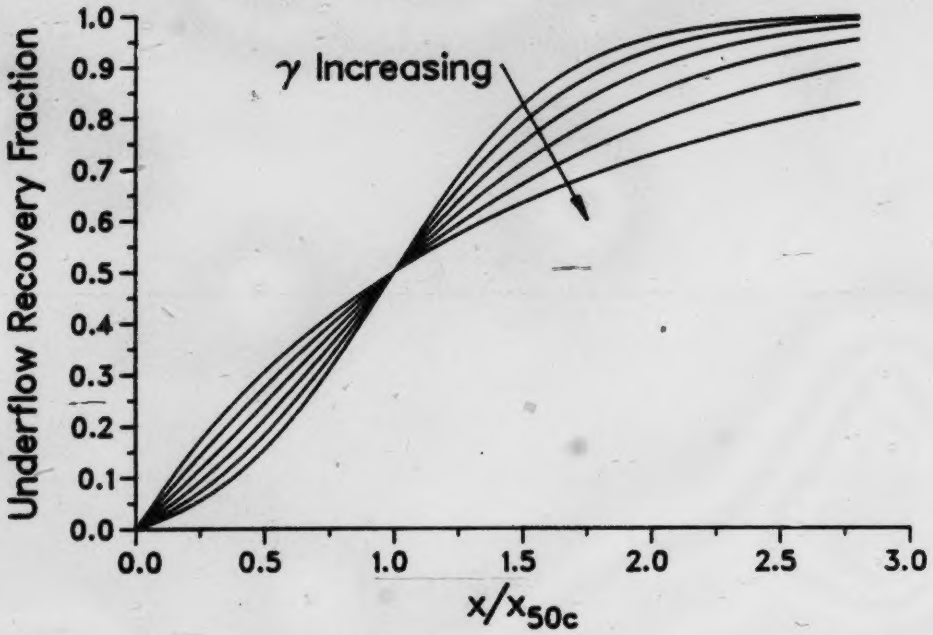


Figure 3.7: The Effect of Variations in c on the Shape of the Reduced Efficiency Curve

hydrocyclone. An empirical expression of the form:

$$\log_{10}(x_{50c}) = K_3 + K_4P + K_5W_o \qquad (3.32)$$

can be used to predict this value. The three constants in the above equation would likely be established by performing a regression analysis on actual hydrocyclone operating data.

It should be noted that equations (3.30) through (3.32) have all been corrected to give particle classification based on centrifugal action alone. In other words, short circuiting of the feed slurry to the underflow stream has been ignored. The actual mass fraction of particles in size class i that report to the underflow stream can be calculated if equation (3.27) is rearranged to give the following equation:

$$Y_{ai} = Rf + (1 - Rf)Y_{ci} \qquad (3.33)$$

where Rf, the mass fraction of feed water recovered in the underflow stream, is given by:

$$Rf = 1 - W_o/W_f \qquad (3.34)$$

It can be seen that the actual efficiency curve described by equation (3.33) will not run through the origin as did the reduced efficiency curve because of the feed stream bypass discussed earlier. After the actual underflow particle size distribution has been determined, material balance constraints provide a simple means of calculating the overflow product stream particle size distribution.

It should be noted that the hydrocyclone model given here can be solved directly for steady state operation when the feed characteristics are known. However, because a pump is normally used to feed the slurry to the hydrocyclone, for dynamic simulations, the pressure flow relation (cf. equation (3.28)) must be solved simultaneously with the corresponding pump pressure flow relation in order to adequately predict all the feed characteristics and thus the transient hydrocyclone behavior.

3.4 Pump and Sump Model

3.4.1 Pump Design

One of the most important pieces of equipment in a grinding circuit, next to mills and hydrocyclones, is the pump that delivers the feed to the hydrocyclone. The performance of this pump is directly related to the recirculating load in a given circuit. The design and especially the sizing of this pump is critical with respect to economic operation of the overall grinding circuit. The efficiency of the pump itself is only a secondary concern because the mills use the majority of the required energy and much of the energy dissipated in the pump is passed to the ore in the form of additional grinding (McElvain and Cave, 1982).

When choosing or designing a pump for comminution service, several considerations are involved, with durability and ease of maintenance not among the least important. Typically, the pump is constructed of abrasion resistant materials with interchangable liners and a second pump is usually available for switching into the circuit when the first requires servicing.

Another important factor in grinding circuit pump design is sizing. The pump must be capable of providing an adequate flow rate along with enough head at the hydrocyclone inlet for proper operation. Care must be taken to size the pump correctly because pumps that are either too large or too small will have poor wear performance due to internal recirculation and increased internal velocities, respectively. As well, light duty pumps with relatively flat pump curves (pressure-flow relations) will cause problems as demonstrated by the pump curve shown as Figure

3.8. The slurry pressure-flow relation is seen to cross the pump curve twice, which creates a region where pump operation will be erratic and cause upsets in the overall circuit performance. This will cause numerical problems in simulation work as well. A steeper pump curve, such as that shown in Figure 3.9, produces a much more consistent behavior and will reduce pump wear and allow for better circuit control. Numerical problems in simulation will also be avoided.

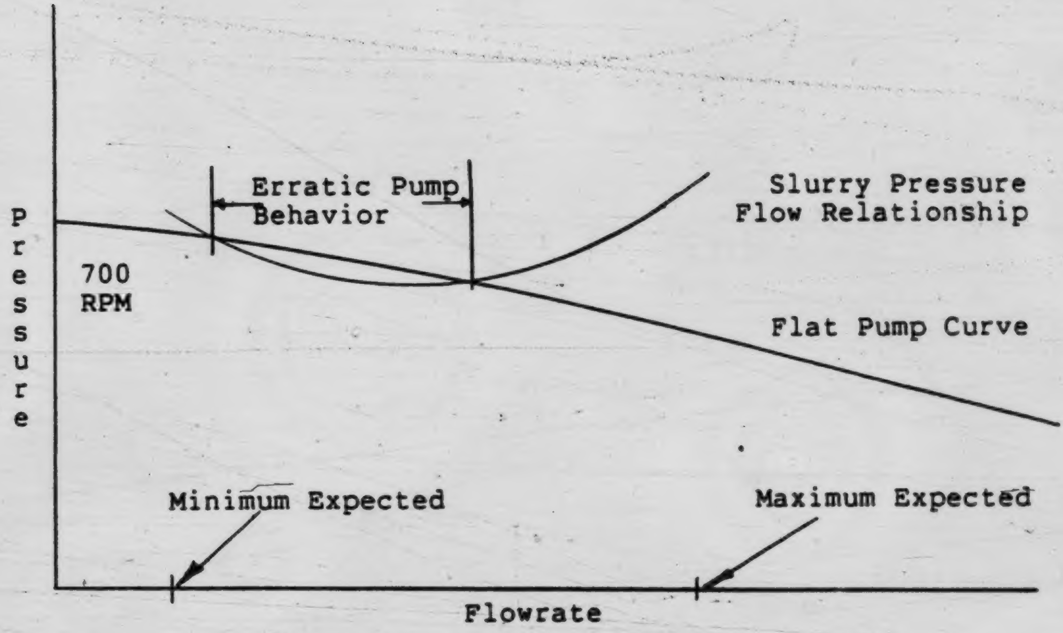

Erratic Pump Behavior
Slurry Pressure Flow Relationship
Pressure
700 RPM
Flat Pump Curve
Minimum Expected
Maximum Expected
Flowrate

Figure 3.8: Flat Pump Curve

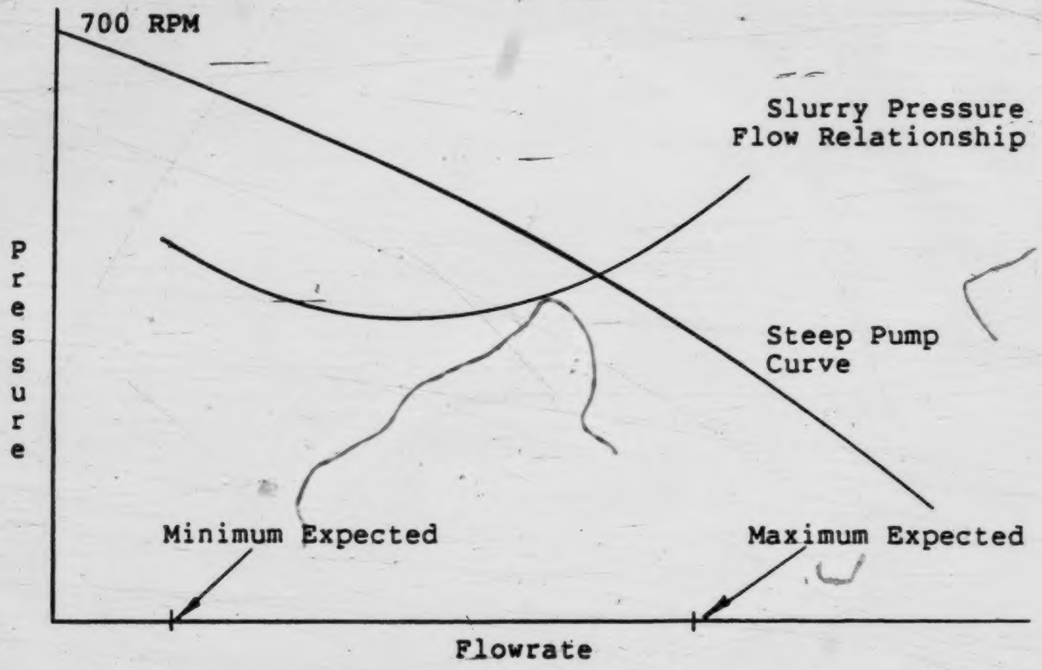


Figure 3.9: Steep Pump Curve

3.4.2 Pump Model

An appropriate pump can usually be chosen from a pump manufacturer's catalogue to give the pressure-flow relationship desired. In simulation work, a mathematical model is required to describe the dynamic head produced by the pump and it is through this model that the hydrocyclone model is linked to the remainder of the grinding circuit. The pump must produce a total dynamic head (TDH) large enough to just overcome the resistance dynamic head (RDH) in the slurry at the hydrocyclone inlet. It should be noted that "dynamic head" is the traditional name given to pressure flow relationships and is not meant to imply time dependence. The pressure heads produced by the pump and that required by the hydrocyclone are dynamic only with respect to flow rate. A simple mechanical energy balance over the pump-cyclone system yields:

$$TDH - RDH = 0 \quad (3.35)$$

The resistance dynamic head consists of several different factors:

1. Net static head from the pump centre line to the cyclone inlet centre line.
2. Pressure head required at the cyclone inlet.
3. Friction head due to pipe and fitting losses.

Each of the above contributions may be calculated if various system variables are known. The net static head contribution (SHC) is given by:

$$SHC = Z - L \quad (3.36)$$

The pressure head contribution (PHC) takes into account the hydrocyclone inlet pressure in metres of pulp as follows:

$$PHC = P/g\rho_p \quad (3.37)$$

The Hazen-Williams equation for friction losses in a slurry pipeline (Lee, 1978) is well known and can be used to determine the friction head contribution (FHC) as follows:

$$FHC = .2083(100/c)^{1.85}(Q_{USGPM}^{1.85}/d_{in.}^{4.8655}) \quad (3.38)$$

where FHC is the head in feet of water per hundred feet of pipe. Assuming steel pipe (c=100) and converting equation (3.38) to SI units gives:

$$FHC = 91.87\rho_p(Q^{1.85}/d_{cm}^{4.8655}) \qquad (3.39)$$

where FCH is now in metres of pulp per 100 metres of pipe. It should be noted that equation (3.39) does not consider fitting losses. This problem can be attacked in a number of different ways. By far the simplest is to use the equivalent pipe length method, adding the appropriate amount of pipe to the length used with equation (3.39).

The resistance dynamic head is thus the summation of equations (3.35), (3.36), and (3.39):

$$RDH = SHC + PHC + FHC \qquad (3.40)$$

The total dynamic head (TDH) of the pump must overcome this resistance in order to satisfy the mechanical energy balance (c.f. equation (3.35)). To model the total dynamic head of the pump, the pump curve as supplied by the manufacturer can be used. Either a quadratic equation of the form:

$$TDH = K_{P1} + K_{P2}Q + K_{P3}Q^2 \qquad (3.41)$$

can be fit to the pump curve, or a digitized version of the curve used in conjunction with an interpolation routine such as FUN1 of DYFLO (Franks, 1972). Interpolation is the simplest procedure; however, the equation representation is more accurate and computationally more efficient.

3.4.3 Sump Model

A sump is usually used to collect the grinding mill product before pumping it to the hydrocyclone for classification. For simulation purposes, it is adequate to model the sump as if it were a perfect mixer. A solids material balance yields the following differential equation:

$$\frac{d(H_s p_i)}{dt} = F_s f_i - P_s p_i \tag{3.42}$$

while a water material balance gives:

$$\frac{dH_w}{dt} = F_w - P_w \tag{3.43}$$

After equation (3.39) is integrated, the total solids holdup is calculated as:

$$S = \sum_{i=1}^{n} H_s p_i \tag{3.44}$$

and the particle size distribution in the sump is found from:

$$p_i = \frac{H_s p_i}{S} \tag{3.45}$$

The level in the sump, assuming constant cross sectional area, is calculated using:

$$\text{level} = (H_s/\rho_s + H_w/\rho_w)/A_x \tag{3.46}$$

3.5 Conveyor and Pipelines

3.5.1 Fixed Time Delays

Conveyors are normally represented as fixed time delays in simulation work. The most common model used is simply a stack or array in computer memory in which elements are selectively placed and removed as simulation time proceeds. The concepts involved are easy to visualize in terms of a "bucket brigade".

A bucket brigade consists of a line of people passing buckets from one to another in series. The first bucket in line is filled while the last one is emptied. The number of buckets being shuffled down the line determines the time is takes for a given bucket to reach the end of the line and be emptied. In terms of a digital computer, the brigade becomes an array or buffer stack while each bucket translates into an element or cell of computer memory inside the buffer as shown in Figure 3.10. It is assumed that it takes no time for the first cell (bucket) to be filled ar the last cell to be emptied. The number of cells is based on the length of the time delay relative to the frequency at which the delayed data are needed.

A more efficient use of the computer memory can be made if the buffer is arranged in a circle as shown in Figure 3.11. Now, rather than shuffling the data from cell to cell the appropriate input and output cells are indexed using a counter. This means that the buffer is being rotated as

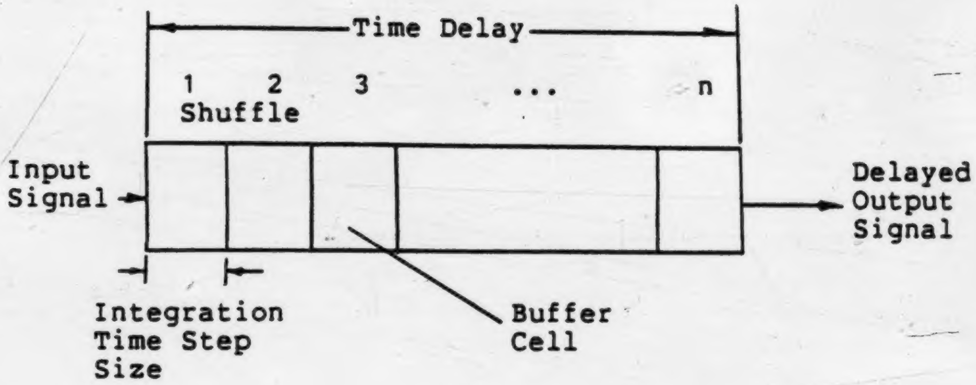

Time Delay
1
2
3
...
n
Shuffle
Input Signal
Delayed Output Signal
Integration Time Step Size
Buffer Cell

Figure 3.10: Computer Representation of a Time Delay

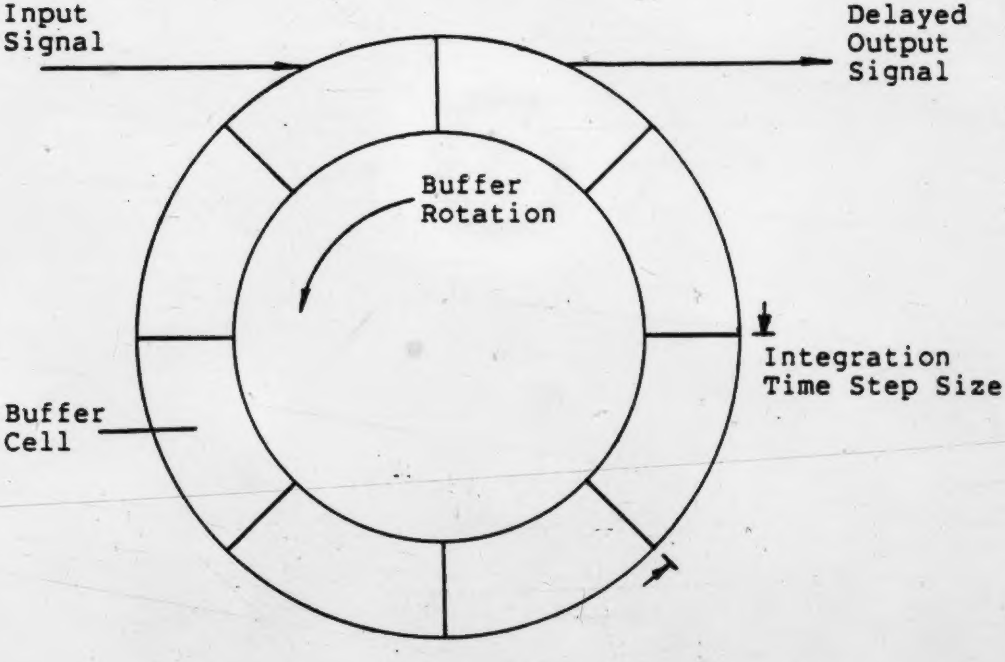

Input Signal
Delayed Output Signal
Buffer Rotation
Integration Time Step Size
Buffer Cell

Figure 3.11: Circular Buffer Schematic

shown by the dotted arrow in the figure. Data is selectively removed and replaced as simulation time proceeds.

3.5.2 Variable Time Delays

Pipelines pose a special problem with respect to modelling and simulation, especially when control system analysis is to be performed. A pipeline is a essentially a variable time delay and as with the fixed time delay, a model not unlike a bucket brigade can be utilized to handle this task. The first bucket is filled by the slurry then its contents are poured into the second bucket in line. The second bucket is not emptied until the first bucket is filled again, and so on. All the attributes of the contents of a given bucket are passed to the next one in line each shuffle. Clearly, the total volume in the time delay is dependent on the number of buckets in the line, while the length of the time delay is dependent on both the number of buckets and how fast the first bucket is filled. The dependence on filling speed is the key to the success of the variable time delay model. It should be noted that the fixed time delay model assumed instantaneous bucket (or block) filling.

The schematic diagram given in Figure 3.12 will further clarify this model. Each of the buckets is represented as a block in this figure. The time delay is composed of N blocks of equal volume, each with an input and an output

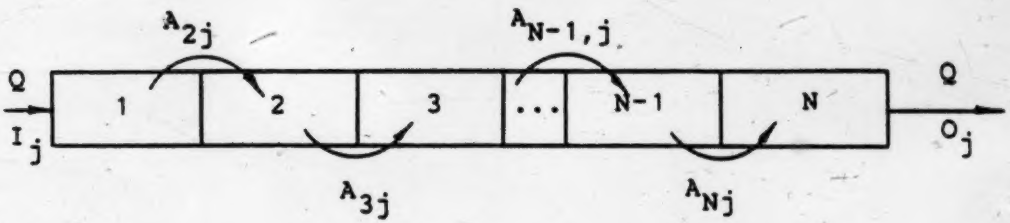


Figure 3.12: Variable Time Delay Schematic Diagram

stream. The total volumetric holdup is constant and all but the first block are assumed to initially be full. The inlet flow is assumed to be incompressible, and even if the inlet flow in a given integration interval is insufficient to fill the first block, an equal volume must leave from the last block in the line. In other words:

$$Q_{in} = Q_{out} \tag{3.48}$$

This also implies that if the volume of the first block is less than the inlet flow in a given integration interval, a fractional amount will remain in the first block. The attributes of each block are shuffled forward only when the accumulated flow entering the variable time delay at least fills the first block. When the inlet flow does not completely fill the first block or a residual amount remains, the outlet flow attributes are determined from a weighted linear interpolation between the attributes of the last two blocks. The weighting factor used depends on how full the first block is and this partially accounts for the continuous nature of the real system.

The following FORTRAN-like algorithm gives an indication of the model formulation for implementation on a digital computer:

```
1. $v = Q\Delta t + v_r$
2. if $v < V/N$ then go to 5
3. $v = v_r = v - V/N$
4. $A_{ij}$ --> $A_{i+1,j}$; $i = N, N-1, \ldots, 1$
   $I_j$ --> $A_{ij}$                              "shuffle"
5. $f = v/(V/N)$
6. $O_j = (1-f)A_{Nj} + fA_{N+1,j}$                "interpolate"
```

The rule used to calculate the number of blocks to be used in the variable time delay model is:

$$2 \le N \le \frac{V}{3Q_m\Delta t} \tag{3.47}$$

Obviously, the larger the value of N, the more closely this model represents a continuous system. This model, although not completely accurate due to discretization, has been found to be a good compromise with respect to simulation speed. It should be further noted that a variable time delay unit can be used in series with the grinding mill model to account for the plug flow component often associated with tumbling mills. This will correct for the residence time distribution without re-estimation of the model parameters of the mill.

3.6 Load Disturbance Modelling

3.6.1 Grinding Circuit Disturbance Inputs

Disturbances to the operation of a mineral grinding circuit can come from several different sources. Perhaps the most important upset in a grinding circuit is a change in the fresh ore feed hardness. Since the hardness of an ore body is generally not uniform, changes in hardness of the ore fed to the grinding mill will occur from time to time thus affecting the particle size distribution in the product stream. Other disturbances result from changes in the feed stream particle size distribution, pressure or flow rate fluctuations in the water supply, and sudden variations in equipment operating performance. The first of these disturbances could occur because of an upset in the primary crushing circuit, while the second can result from water demand changes in the rest of the plant. The last disturbance mentioned above can result from mechanical failures such as rod breakage or pump drive belt breakage.

In this work, only hardness changes have been considered because, in general, this will be one of the more common disturbances in a typical grinding operation and also has a large effect on the final product quality.

3.6.2 Simulation of Ore Hardness Changes

Ore hardness is reflected in the values chosen for the selection function, k, with harder ore particles breaking less frequently than softer particles. Normally, changes in hardness appear as load disturbances to the system. At present, there is no means of measuring ore hardness on line, however, a Kalman filter approach can be used to provide an estimate. This is beyond the scope of the current work.

Changes in ore hardness are modelled by changes in the selection function of the grinding mill feed material and in general, there are two approaches that can be used for this purpose. The first makes use of a multiplication factor (constant or variable) to alter the selection rate function while the second assumes that the feed stream is composed of two different ore components with different selection rate functions (hardness) present in differing amounts. A change in hardness appears as a reduction in concentration of one component or the other.

The multiplier method is the simplest technique to use and, as implied above, can be implemented in two different ways. The easiest is to treat the multiplier as a constant and simply multiply the selection rate function by the constant to obtain a new selection rate function for the ore. This has the effect of instantaneously changing the breakage rate in the mill and immediately affecting the mill performance which is not true in practice. A second, more realistic, implementation is to make the multiplier into a slurry attribute.

A mass balance around the mill, assuming the multiplier to be a slurry attribute and perfect mixing in the mill, gives:

$$\frac{dH_sM_p}{dt} = F_sM_f - P_sM_p \tag{3.49}$$

where M_f is the multiplier associated with the feed stream and M_p is the multiplier associated with the product stream or holdup. Because the mill volumetric holdup (V) and the solid to liquid feed mass ratio are constant, the solid mass holdup of the mill is constant and the solid feed/product mass flow rates are equal. Thus, dividing equation (3.45) by the solids holdup yields:

$$\frac{dM_p}{dt} = \frac{M_f - M_p}{\tau} \tag{3.50}$$

and the selection function for the mill contents is adjusted by the factor, M_p. It can be seen that this multiplication factor is time dependent and will affect the dynamic behavior of the mill.

The other technique for modelling ore hardness changes, as mentioned earlier, involves assuming that the feed stream consists of two different ore components with identical properties except for the selection function. Using this approach, the mill model for component l of the feed stream follows from an unsteady state material balance similar to that performed in the derivation of the earlier dynamic mill model (cf. equation (3.18)):

$$\frac{d(H_l p_{il})}{dt} = F_l f_{il} - P_l p_{il} - k_{il} H_l p_{il} + \sum_{j=1}^{i-1} b_{ij} H_l k_{jl} p_{jl} \tag{3.51}$$

where:

$$F_l = Qc_{fl} \tag{3.52}$$

and:

$$P_l = Qc_{pl} \tag{3.53}$$

Expanding the left hand side of equation (3.51) gives:

$$\frac{d(H_l p_{il})}{dt} = H_l \frac{dp_{il}}{dt} + p_{il} \frac{dH_l}{dt} \tag{3.54}$$

The mass holdup of each component can be calculated from:

$$H_l = Vc_{pl} \tag{3.55}$$

and the derivative of equation (3.55) with respect to time is:

$$\frac{dH_l}{dt} = \frac{d(Vc_{pl})}{dt} \tag{3.56}$$

However, noting that the volumetric holdup of the mill is assumed constant (cf. Section 3.2), equation (3.56) becomes:

$$\frac{dH_1}{dt} = V\frac{dc_{p1}}{dt} \tag{3.57}$$

A concentration balance for component 1 around the mill gives:

$$\frac{Vdc_{p1}}{dt} = Q(c_{f1} - c_{p1}) \tag{3.58}$$

This equation is the key to the success of this method of modelling hardness changes. Unlike the original model where changes in concentration of the solid material in the feed stream were ignored, the changes in the concentration of a particular solid component reflected by equation (3.58) result in a change in overall ore hardness. Using equations (3.52) through (3.58) with equation (3.51), it follows that:

$$p_{i1}Q(c_{f1}-c_{p1})+H_1\frac{dp_{i1}}{dt}=f_{i1}Qc_{f1}-p_{i1}Qc_{p1}$$
$$-k_{i1}H_1p_{i1}+\sum_{j=1}^{i-1}b_{ij}H_1k_{j1}p_{j1} \tag{3.59}$$

Collecting the terms in Q and dividing through by the solids mass holdup of the mill yields:

$$\frac{dp_{i1}}{dt} = (f_{i1}-p_{i1})\frac{Qc_{f1}}{Vc_{p1}}-k_{i1}p_{i1}+\sum_{j=1}^{i-1} b_{ij}k_{j1}p_{j1} \tag{3.60}$$

with the overall particle size distribution given by:

$$p_i = \frac{c_{p1}p_{i1}}{\Sigma c_{p1}} \tag{3.61}$$

Although this technique involves more computational effort, it proves to be more general than the previous method and is perhaps the most realistic.

4. Grinding Circuit Simulator Description

4.1 Introduction

In simulation work, computer software is a very critical part of the researcher's "tool box". This tool relieves the tedium associated with the lengthy computations normally involved as well as allowing various system parameters to be changed easily and new results quickly generated. The purpose of this chapter is twofold. The primary function is to describe the simulation software package, MINSIM, developed for studying the dynamic behavior of mineral grinding circuits at the University of Alberta. As well, this chapter can serve as both a programmer's and user's manual. The first time user may wish to skip directly to Section 4.4, where "quick start" instructions for MINSIM are given.

First, it is necessary to choose an appropriate simulation language or technique which will provide a strong basis for the development of a mineral grinding circuit simulator. Several factors are involved in this choice including cost, ease of use and understanding, as well as portability. As one of the primary objectives of this work is to demonstrate the usefulness of a grinding circuit simulator to the mineral plant engineer for both operator training and control scheme design purposes, the DYFLO simulation program with its library of FORTRAN IV subroutines becomes attractive (Franks, 1972). This simulation program provides a well documented process unit operation oriented executive framework and programming structure on which Wood and Flintoff (1982) have shown a dynamic mineral processing simulator can be based. Another advantage in the choice of DYFLO is that several utility routines to handle standard functions such as integration and data printing already exist. As well, the fact that DYFLO is based on a well known and widely available computer language, FORTRAN IV, implementation of such a system in the plant environment would not be expected to be any problem.

The implementation of the original grinding circuit simulator has been documented by Wong (1984), and Flintoff et al. (1985) have demonstrated its use for open loop and single variable control. In this work, the simulation package has been upgraded to take advantage of the stiff integration routines supplied in DYFLO2 (Franks, 1982) and extensions have been made to allow for the investigation of advanced multivariable control strategies. The simulation package currently consists of three general subroutine libraries and two executive programs.

The next section of this chapter provides a description of DYFLO2.LIB, the DYFLO2 library of subroutines; MIN.LIB, the mineral grinding circuit unit operation subroutine library, and STC.LIB, the self-tuning controller subroutine library. The structure of the executive programs developed for the simulation studies presented in Chapters 5 and 6 is discussed in Section 4.3. The interactive user interface developed for the Lake Dufault Circuit is described in more detail in Section 4.4 along with general execution instructions. This section essentially contains all that is needed to use the system and can be considered to be the user's manual. This chapter concludes with instructions for adding new modules or flowsheets to the simulator in Section 4.5 and will be of interest to programmers.

4.2 Simulation Language Description

4.2.1 The DYFLO2 Subroutine Library

The DYFLO2 simulation language was chosen for this work due to its availability and ease of upgrading to include more advanced capabilities. As noted earlier, the DYFLO2 subroutine library, DYFLO2.LIB, is an extension of DYFLO which has been in use at the University of Alberta in the Department of Chemical Engineering since the mid 1970's. Documentation for the original version of DYFLO exists in the book by Franks (1972) and for DYFLO2, an updated manual has been provided (Franks, 1982). Subroutine modules that model standard chemical processing unit operations such as heat exchangers, mixing tanks, and distillation columns are available in DYFLO2, along with subroutines for first and second order transfer functions and continuous P, PI, and PID controllers for use in control studies. Utility subroutines capable of simulating time delays, arbitrary functions and performing explicit and implicit first, second, and fourth order integration also are available. A complete list of the DYFLO2 subroutines used in this work, their functions, and the interface requirements is given in Appendix A. Verification of the correct operability of some of these routines is presented in Appendix B, while Wong (1984) provides verification for the control related subroutines. A brief review of numerical integration techniques and how they are implemented in DYFLO2 is given in Appendix C.

As mentioned earlier, several extensions to the original DYFLO2 program have been made, especially in the area of control algorithms and strategies. Table 4.1 summarizes the new subroutines and their functions. A detailed description of each of these subroutines, including data requirements and usage, can be found in Appendix D. Verification of the correct operability of the lead lag compensator and the time delay compensator routines are presented in Appendix E and Appendix F, respectively.

4.2.2 The Mineral Grinding Circuit Unit Operation Subroutine Library

Several of the mineral grinding circuit unit operation models discussed in Chapter 3 were originally implemented by Flintoff and Wood (1982) as FORTRAN IV subroutines using the DYFLO approach. In this work, these routines have been upgraded to make use of the implicit integration routines available in DYFLO2, and currently exist as the subroutine

Table 4.1: DYFLO2.LIB New Subroutines and Functions

Subroutine	Function
CON2X2	Continuous two input-two output controller with any combination of P, PI, or PID available.
DISPID	Discrete velocity form of P, PI, or PID controller with derivative action only on the process output signal.
DPID2	Discrete velocity form of P, PI, or PID implementation with error actuated control action
INTCPD	Highly coupled ODE recursive integration algorithm.
I2BY2	Interacting two by two first order transfer function model including time delay and load capabilities.
JFUNC	Performs function evaluations for NUMJAC. This subroutine requires the support of the user written routines TKCALC and FFCALC to compute the problem dependent reciprocal time constants and forcing functions.
LLCDER	Lead lag compensator using the numerical derivative implementation.
LLCOMP	Lead lag compensator implementation using only integration.
NUMJAC	Computes the numerical Jacobian matrix of a system of ordinary differential equations.
PTRAIN	Square wave pulse train generator.

Table 4.1 continued.

RATCON	Ratio controller without dynamics.
RELAX	Smart relaxation factor calculation for use with INTCPD.
TDC	Multivariable time delay compensator using the technique proposed by Ogunnaike and Ray.
ZOH	Zero order hold function.

library, MIN.LIB. It should be noted that due to the highly coupled and time varying nature of the set of differential equations that are used in modelling the grinding mill, it is not possible to utilize the implicit integration routine directly. In the implicit formulation of the mill model, the forcing functions are coupled through the product stream particle frequency size distribution. As well, the forcing function along with the reciprocal time constant are functions of the mill time constant and the selection function, both of which vary with time. Franks (1982) suggests that for situations such as this, a recursive solution procedure, involving the calculation and inversion of the Jacobian matrix, should be used as discussed in Appendix C. A new integration routine based on this theory was developed and a description of this software can be found in Appendix D. An example problem showing the analytical and numerical solutions to a set of "stiff" differential equations is presented in Appendix B as verification of the software discussed in Appendix D. The results of this example suggest that the additional computational requirements of the recursive integration technique outweigh any benefit gained by the ability to use a larger integration time step. This is especially true when the loss of solution accuracy is considered due to single precision computer operation to maintain compatability with the original DYFLO2 software. Implementation of the entire simulator in extended precision to gain solution accuracy is felt unjustified in view of the increased computational costs that would be encountered. A simpler and more cost effective integration technique is the predictor corrector formulation. This technique uses the explicit method to predict the solution and then the implicit method is used to correct it. This has the effect of reintroducing the integration time step size limitations imposed by the explicit integration method to a certain extent, but the solution accuracy is not compromised when a slightly larger time step is used and some computational cost benefits are realized. Therefore, this technique has been incorporated into the appropriate unit operation subroutines. It should be noted that although the code allows up to fourth order integration to be used, first order integration was used exclusively in this work to further minimize computing costs. Table 4.2 summarizes the mineral grinding circuit subroutines and briefly presents the function of each. More details on the data requirements and subroutine functions can be found in Appendix A.

4.2.3 Self-Tuning Controller Subroutine Library

The subroutine library, STC.LIB, is based on the multivariable self tuning controller discussed in Chapter 2 (cf. Section 2.7). This subroutine library was developed from software used by Langman (1987). The original software was adapted for compatability with the other grinding circuit simulator system libraries by developing a DYFLO-STC interface routine and a separate self-tuning controller initialization subroutine. The original code was not altered with the exception of the routine, MIMOIN, which was rewritten to read input data from an ASCII file. In addition, the FORTRAN COMMON block, STC, was added to tranfer data to and from the self-tuning controller and several unnecessary COMMON blocks were deleted. Table 4.3 sumarizes the various subroutines in this library, along with a brief outline of the function of each. Further details on the interface requirements of each routine can be found in Appendix A.

4.3 Simulator Structure

The framework on which the executive program structure is based, follows from the DYFLO programming technique of Franks (1972). This programming style dictates that the program be broken into four sections, each performing a

Table 4.2: MIN.LIB Subroutines and Functions

Subroutine	Function
BLEND	Blends all the attributes of up to three input streams into one output stream.
CLASS	Performs the calculations associated with the hydrocyclone classifier model.
CONVEY	Simulates a fixed time delay for a process stream defined in a STREAM data structure.
MILL	Performs the calculations associated with the rod or ball grinding mill models.
MILOUT	Writes grinding mill physical data to a simulation output data file.
PLUGFL	Simulates a variable time delay for a process stream defined in a STREAM data structure.
SMPPMP	Performs the calculations associated with the sump and pump models.
STROUT	Writes STREAM data structure to a simulation output data file.
VOLCNT	Adjusts the sump water feed rate to maintain a constant volumetric flow rate in the hydrocyclone feed stream.

Table 4.3: STC.LIB Subroutines and Functions

Subroutine	Function
CNTR4I	Self-tuning controller initialization.
CONTRL	The self-tuning controller executive program for a two input-two output system with capabilities to handle measurable disturbance inputs also. This routine can also be used as a standard discrete P, PI, or PID controller with the option of actuating the self-tuning control action at a prespecified time.
CONTR4	The self tuning controller routine called from the DYFLO2 program, converts the input/output data for compatability with CONTRL. This is the DYFLO-STC interface program.
IDENT1	Performs recursive least squares parameter estimation calculations.
IDENT2	Performs recursive least squares parameter estimation calculations with square root factorization.
IDENT3	Performs recursive least squares parameter estimation calculations with upper diagonal factorization.
IDENT4	Performs recursive learning parameter estimation calculations.
MIMOIN	Self-tuning controller initialization, reads data from the file STC.D.
MOIST1	Self-tuning controller calculations.

Table 4.3 continued.

MOIST2	Self-tuning controller calculations.
PWRITE	Writes covariance matrix to an output data file.
ULIMIT	Limits the change in control action per control (or sampling) interval.

specific function. The sections and the functions performed in each section are given in Table 4.4. This information in terms of a data flow diagram is repeated in Figure 4.1. Each bubble in this diagram represents the specific function performed by each section of the program. The particular data that is transfered between each section of the program is described by the captions associated with the lines connecting the bubbles and the arrowheads indicate the direction in which data flows. The flow chart in Figure 4.2 illustrates this in terms of the logic used in the executive. Printing of all output data is handled in the derivative section as is the logic for deciding when execution of the program is to terminate. The integration section advances the simulation time by an amount equal to the integration time step each complete integration pass. A complete integration pass, as defined by Franks (1972), depends on the integration order. First order integration causes the program to cycle through the derivative and integration sections only once per integration time step,

Table 4.4: DYFLO Executive Program Structure and Function Summary

Section	Functions Performed
Initialization	All constants are specified or calculated and initial values assigned for all variables, and the simulation output headings are printed. The section executes only once at the begining of the simulation.
Derivative	All derivative values and any other calculations (except integration) that must be computed every time step are performed here. Simulation output printing and a test for completion of the simulation conclude this section.
Integration	All integration calculations are performed and the routine, INTI, is always the first executable line in this section of the program. This means that in general~~,~~ any routine that contains a call to an integration routine is called from this section. The exception to this rule is for cases where the logical variable LSTR is used in the process routine (eg. HLDP) to control the integration calculations. Program execution cycles to the start of the derivative section at the conclusion of this section.
Termination	All terminal calculations that must be performed only once at the end of the simulation reside in this section. This section concludes the simulation program.

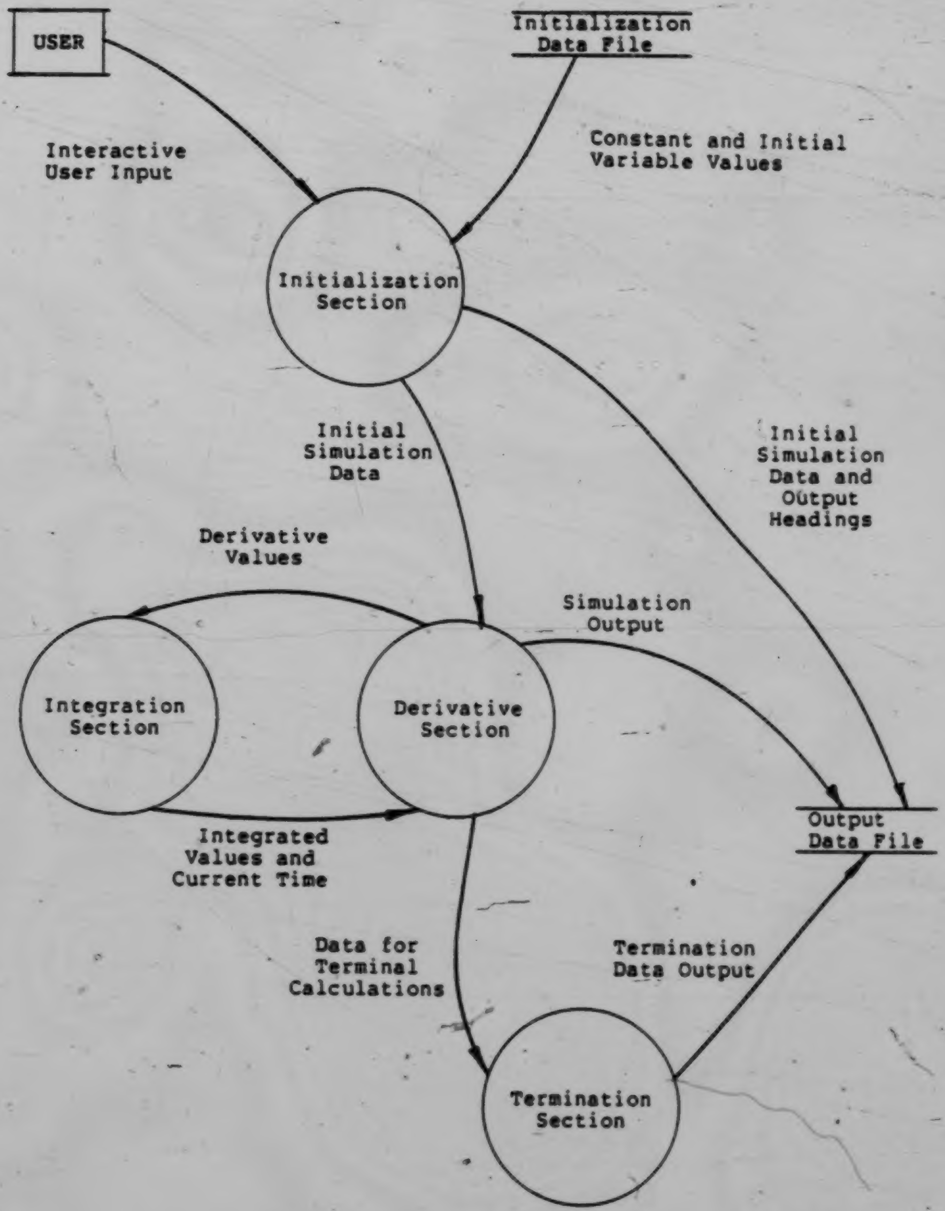

USER
Initialization Data File
Interactive User Input
Constant and Initial Variable Values
Initialization Section
Initial Simulation Data
Initial Simulation Data and Output Headings
Derivative Values
Simulation Output
Integration Section
Derivative Section
Output Data File
Integrated Values and Current Time
Data for Terminal Calculations
Termination Data Output
Termination Section

Figure 4.1: A Data Flow Diagram for a General DYFLO2 Executive Program

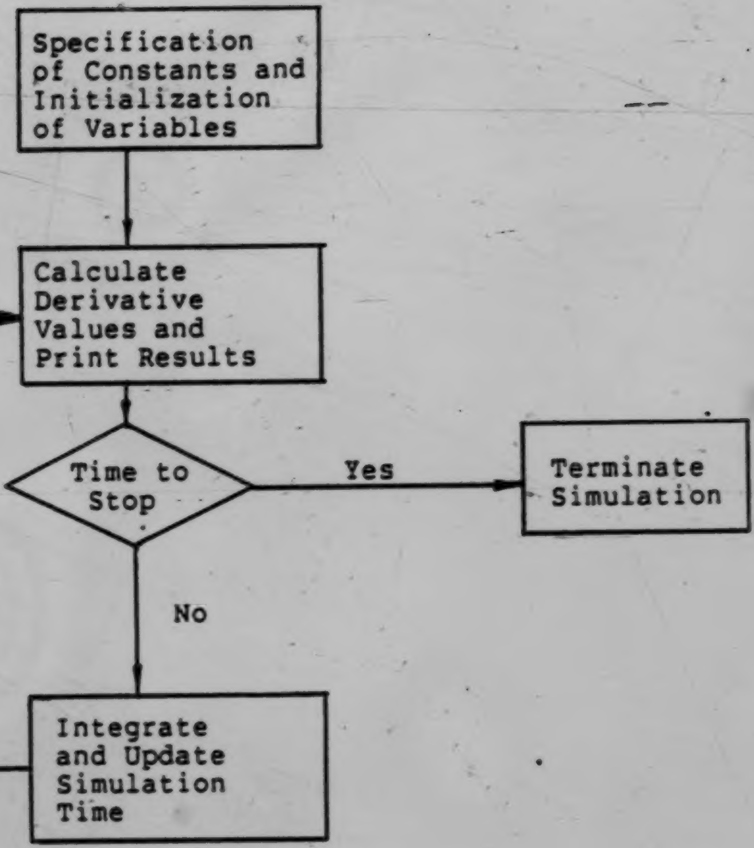

Specification of Constants and Initialization of Variables
Calculate Derivative Values and Print Results
Time to Stop
Yes
Terminate Simulation
No
Integrate and Update Simulation Time

Figure 4.2: A Program Flow Chart for a General DYFLO2 Executive Program

while second and fourth order integration causes the program to cycle two and four times, respectively. Thus, first order itegration results in a single pass every interation, while the second and fourth order integration techniques require multiple passes for each integration interval. Additional details on the integration scheme used in DYFLO2 can be found in the discussion given in Appendix C.

4.3.1 The Closed Circuit Grinding Simulator

The Closed Circuit Grinding Simulator is a menu driven software system that has been developed as part of MINSIM and is intended to demonstrate many of the capabilities of the MINSIM simulation package. The subroutine libraries discussed in Section 4.2 provide the basis of the simulator, while an executive structure is used to tie the system together. It should be noted that the closed circuit simulator was used only with the Lake Dufault circuit. Future work includes the simulation of other circuits. All subsequent discussions pertaining to this simulator apply in a general sense to the MINSIM software package unless otherwise noted.

In this case, a two tiered executive structure has been developed. The first level or tier is used to perform the interactive duties of the simulator and can be considered to be an extension of the initialization and termination sections of the second level executive. The second level executive is in fact a set of executive programs which perform the actual simulation calculations and are of the "standard" DYFLO structure discussed earlier. Each of these executives are specific to the grinding circuit control configuration used and the program logic follows that shown in Figure 4.2. It should be noted that the second level executives are essentially transparent to the user, as it appears to the user that the level one executive performs the complete simulation. This type of executive structure offers several advantages. Foremost, it allows new second level executive programs to be developed and tested independently for subsequent incorporation into the first level executive as a subroutine. As well, duplication of the interactive "front end" is avoided.

The functions performed by the first level executive are shown in the data flow diagram in Figure 4.3 where it can be seen that this executive is responsible for interacting with the user to obtain global simulation data and writing it to the output file. Additional details of the interactive nature of this executive, along with a description of the data the user is expected to input to the system, is discussed in Section 4.4 of this chapter.

The second level executive is broken into its functional components in Figure 4.4. This executive level consists of five parts, only one of which executes during a given simulation run. The global simulation data entered by the user is passed through the first level executive program to

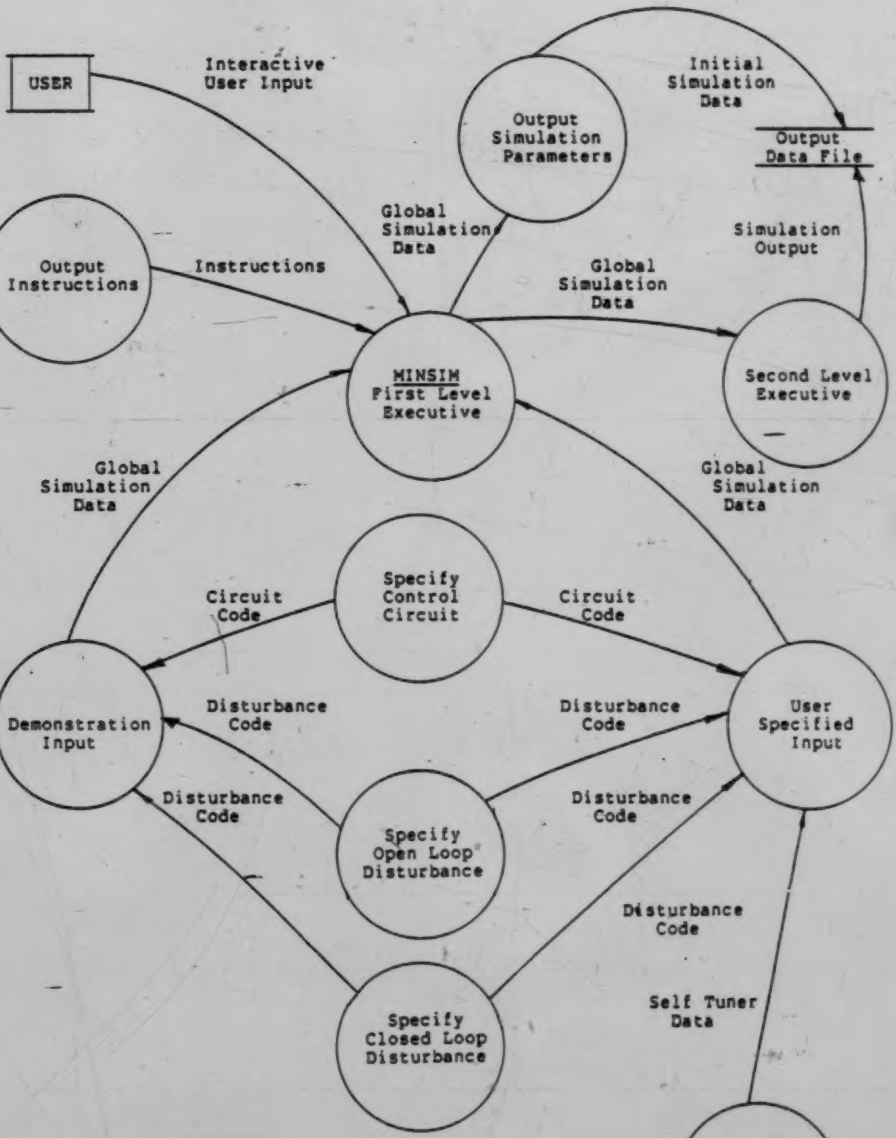

USER
Interactive User Input
Output Simulation Parameters
Initial Simulation Data
Output Data File
Output Instructions
Instructions
Global Simulation Data
Global Simulation Data
Simulation Output
MINSIM First Level Executive
Second Level Executive
Global Simulation Data
Global Simulation Data
Specify Control Circuit
Circuit Code
Circuit Code
Demonstration Input
Disturbance Code
Disturbance Code
User Specified Input
Disturbance Code
Disturbance Code
Specify Open Loop Disturbance
Disturbance Code
Self Tuner Data
Specify Closed Loop Disturbance
Specify Self-Tuner Parameters

Figure 4.3: First Level Data Flow Diagram for the Closed Circuit Grinding Simulator

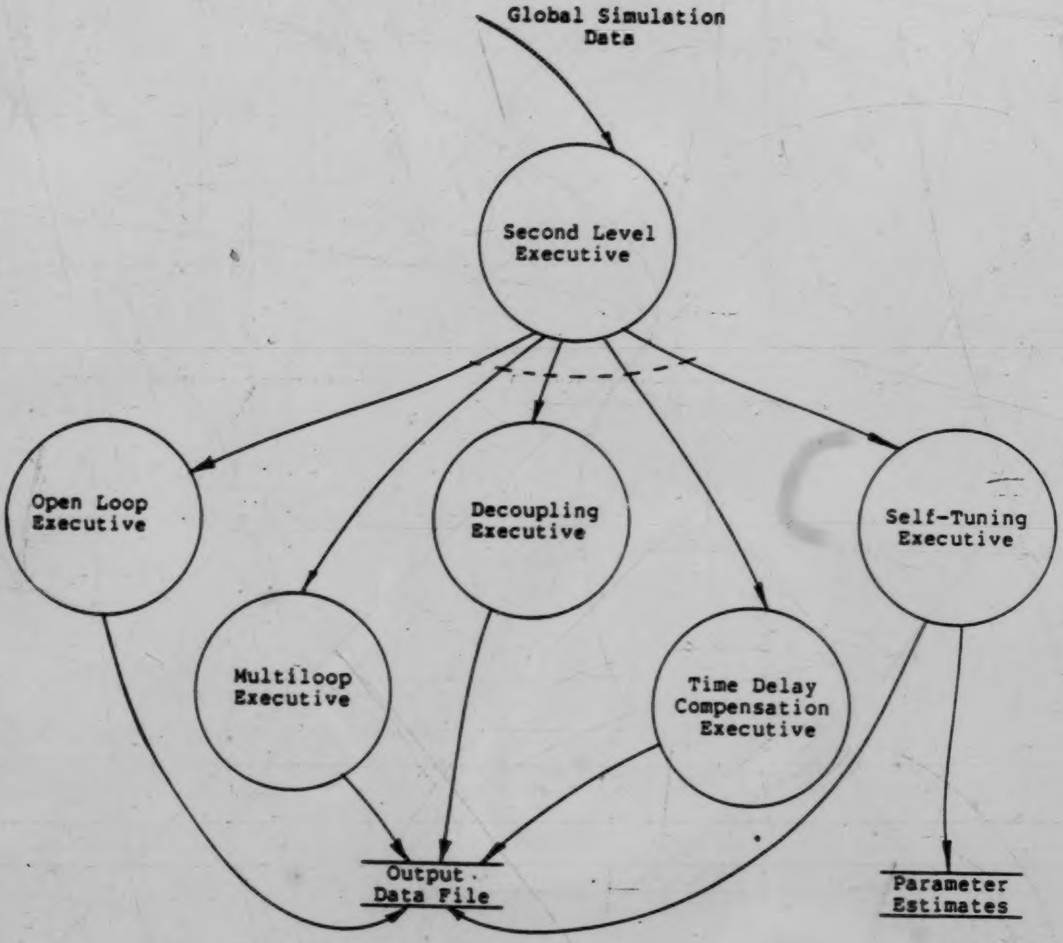

Global Simulation Data
Second Level Executive
Open Loop Executive
Decoupling Executive
Self-Tuning Executive
Multiloop Executive
Time Delay Compensation Executive
Output Data File
Parameter Estimates

Figure 4.4: Second Level Data Flow Diagram for the Closed Circuit Grinding Simulator

the second level, and the simulation calculations are performed. It should be noted that dashed line crossing various bubble connections in this figure indicates that only one of the functions connected in this way are performed.

To gain a better appreciation of the overall implementation of the closed circuit simulator, a program structure diagram is provided in Figure 4.5. The functions performed by the simulator are represented as blocks in this figure, named after the subroutine used to accomplish the particular function. Table 4.5 summarizes these subroutines and their functions. The lines connecting the blocks give an indication as to the subroutine linking within the first level executive of the simulator. This main executive has been given the title MINSIM, and the interactive user input subroutines used by this executive are shown below to represent the hiearchy of the system. It is interesting to note that a significant advantage of this form of representation for a software system is that the order of execution corresponds to the position of the subroutine block, from left to right. As can be observed from the program structure diagram in Figure 4.5, the second level executive programs are called as subroutines from the first level executive program after the initialization subroutines have completed their functions. It should be noted here that the diamond symbol used to connect several of the subroutine blocks in the diagram in Figure 4.5 is meant to

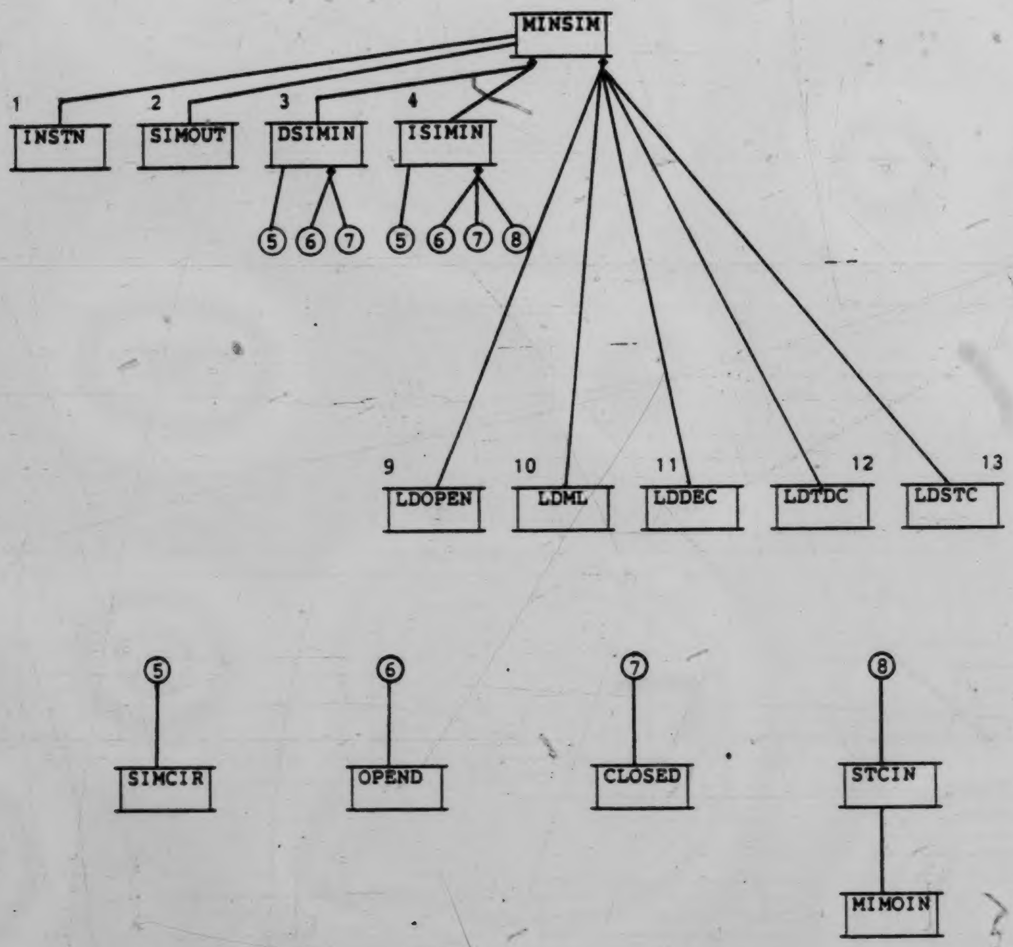

MINSIM
1
INSTN
2
SIMOUT
3
DSIMIN
4
ISIMIN
5
6
7
5
6
7
8
9
LDOPEN
10
LDML
11
LDDEC
12
LDTDC
13
LDSTC
5
SIMCIR
6
OPEND
7
CLOSED
8
STCIN
MIMOIN

Figure 4.5: Program Structure Diagram for the Main Executive of the Closed Circuit Grinding Simulator

Table 4.5: The Closed Circuit Simulator Subroutines and Functions for the First Level Executive Program

Subroutine	Function
MINSIM	First level executive program. This program schedules the interactive user input to the simulation run and calls the appropriate second level executive to perform the simulation calculations. Control is returned to this program after the termination section of the second level executive is finished to conclude the simulation.
INSTN	This subroutine provides the first time user with some brief online instructions on the use and features of the simulator. Simulation cost warnings are also presented to the user in this routine.
DSIMIN	This subroutine allows the user to set up the simulator to run in the demonstration mode. It allows the user to choose the control system to be simulated and the disturbance to be used. The controller settings, disturbance magnitude, and all other global simulation parameters are selected automatically by this routine.
ISIMIN	This subroutine allows the user to specify all the important simulation parameters such as the controller settings, the type of grinding circuit model to be used, and the duration of the simulation. This requires that the user be reasonably familiar with the operation of the Lake Dufault circuit, although typical input examples are provided as an aid. It should be noted that the self-tuning controller option is available only through this subroutine.

Table 4.5 continued.

SIMCIR	This subroutine constructs the control system configuration choice menu. It is responsible for interacting with the user to obtain the circuit code. Error checking is performed to validate the option choice.
OPEND	This subroutine constructs the open loop control system configuration disturbance input menu. It is responsible for interacting with the user to obtain the disturbance code. Error checking is performed to validate the option choice.
CLOSED	This subroutine functions in a similar fashion to OPEND, except the closed loop disturbance input menu is constructed.
STCIN	This subroutine is used if the self-tuning controller option is used. It is responsible for interacting with the user to obtain the disturbance input to the system along with some discrete controller parameters such as sampling time.
LDOPEN	The second level executive program responsible for simulating the open loop response of the Lake Dufault grinding circuit.
LDML	The second level executive program responsible for simulating the Type I or Type II multiloop control system behavior of the Lake Dufault grinding circuit.
LDDEC	The second level executive program responsible for simulating the Type I or Type II noninteracting control system behavior of the Lake Dufault grinding circuit.

Table 4.5 continued.

LDTDC	The second level executive program responsible for simulating the Type I or Type II time delay compensated control system of the Lake Dufault grinding circuit.
LDSTC	The second level executive program responsible for simulating the Type I or Type II self tuning control system behavior of the Lake Dufault grinding circuit.

indicate that a choice is made in the upper level routine as to which one of the lower level subroutines is called, which in turn implies that only one of the lower routines is executed in a given simulation run. An example will help to clarify the features of this type of program structure diagram. The subroutines at the ends of the interfaces labelled 1 and 2 in the diagram in Figure 4.5 are both used in a given simulation, while only one of the subroutines, DSIMIN or ISIMIN, will be called. Furthermore, SIMCIR is called from both of these upper level routines and a decision must again be made as to which of the functions 6 and 7 (from DSIMIN), or 6, 7 and 8 (from ISIMIN), will be performed. After returning from 3 or 4, the MINSIM executive program must decide on the second level executive subroutine to be used. These second level executive subroutines appear at the termination of the interfaces labelled 9 through 13.

The program structure diagram in Figure 4.6 presents a detailed breakdown of the operations involved in the open loop simulation executive program which follows the classic

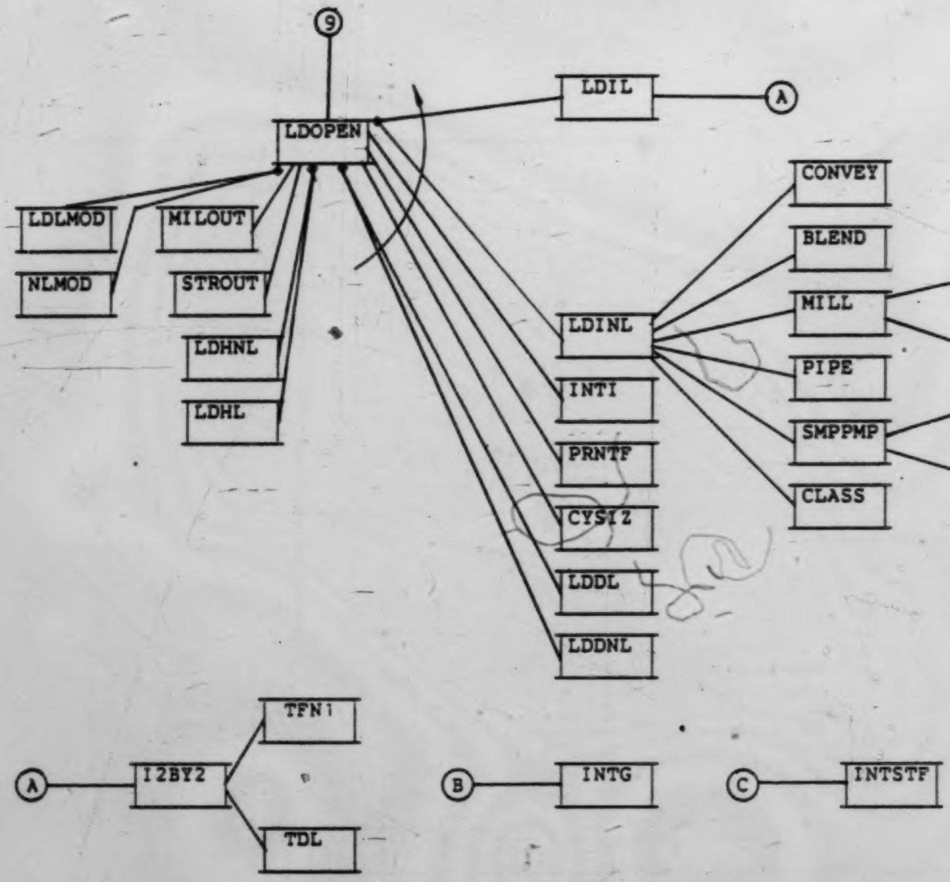

9
LDOPEN
LDIL
A
LDLMOD
NLMOD
MILOUT
STROUT
LDHNL
LDHL
LDINL
INTI
PRNTF
CYSIZ
LDDL
LDDNL
CONVEY
BLEND
MILL
PIPE
SMPPMP
CLASS
B
C
B
C
A
I2BY2
TFN1
TDL
B
INTG
C
INTSTF

Figure 4.6: Program Structure Diagram for the Open Loop Executive of the Closed Circuit Grinding Simulator

DYFLO2 executive structure outlined earlier. The subroutines that appear in this figure that are specific to the Lake Dufault circuit are summarized in Table 4.6. It should be noted that the cyclical nature of the derivative and integration sections of the executive are indicated in Figure 4.6 by the arrow crossing the appropriate interface connections.

Figures 4.7 through 4.10 present program structure diagrams for the other four executive programs of the closed circuit simulator. As can be seen from these figures, the structure of each is similar. The only differences are in the integration sections where additional or different control routines may be used. All the subroutines in these figures that are specific to the Lake Dufault circuit are summarized in Table 4.6, along with a brief functional description.

4.3.2 Open Circuit Grinding Simulator

The open circuit grinding simulator is a menu driven software system that has been developed as part of the MINSIM system and is intended to demonstrate some additional features and capabilities of the MINSIM simulation package that were not included in the closed circuit simulator discussed in the previous section. As with the closed circuit simulator, the subroutine libraries discussed in Section 4.2 provide the basis of the simulator and an executive structure is used to tie the system together.

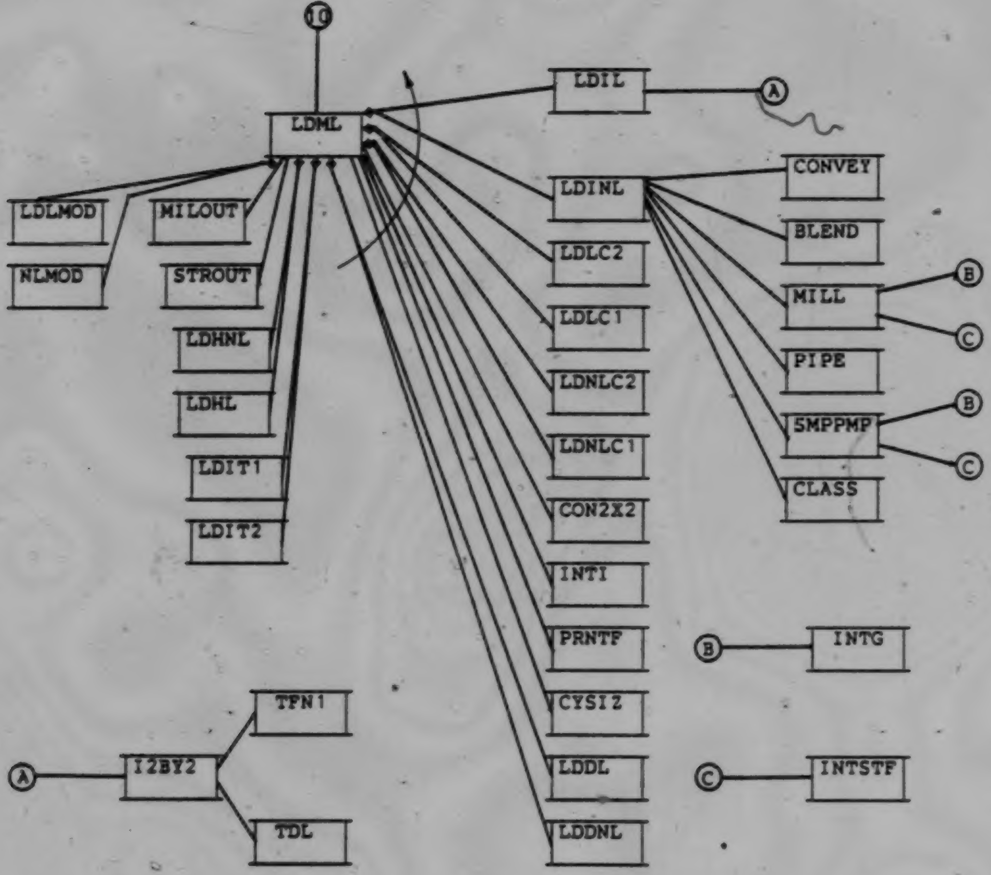

10
LDML
LDIL
A
LDINL
CONVEY
BLEND
MILL
B
C
PIPE
SMPPMP
B
C
CLASS
LDLMOD
MILOUT
NLMOD
STROUT
LDHNL
LDHL
LDIT1
LDIT2
LDLC2
LDLC1
LDNLC2
LDNLC1
CON2X2
INTI
PRNTF
CYSIZ
LDDL
LDDNL
B
INTG
C
INTSTF
A
I2BY2
TFN1
TDL

Figure 4.7: Program Structure Diagram for the Multiloop Type I and Type II Control Scheme Executive for the Closed Circuit Grinding Simulator

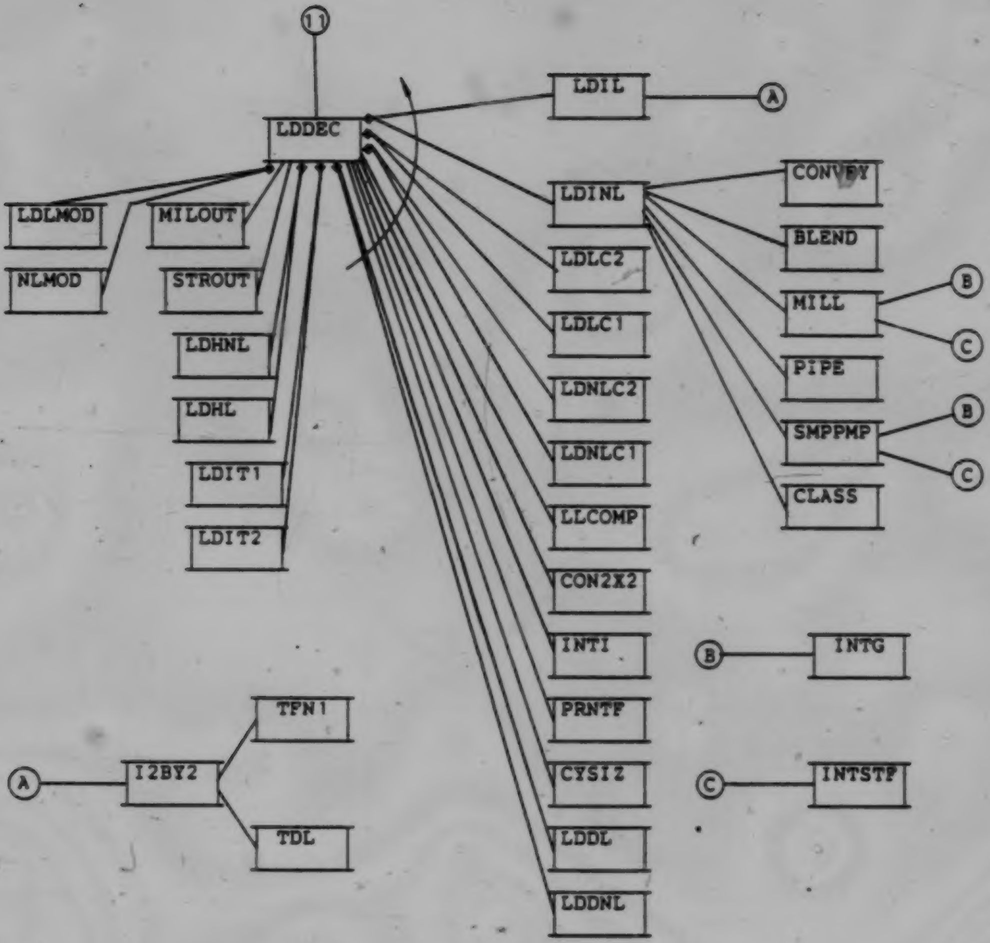

11
LDDEC
LDIL
A
LDLMOD
MILOUT
NLMOD
STROUT
LDHNL
LDHL
LDIT1
LDIT2
LDINL
CONVEY
BLEND
MILL
B
C
PIPE
SMPPMP
B
C
CLASS
LDLC2
LDLC1
LDNLC2
LDNLC1
LLCOMP
CON2X2
INTI
PRNTF
CYSIZ
LDDL
LDDNL
B
INTG
C
INTSTF
A
I2BY2
TFN1
TDL

Figure 4.8: Program Structure Diagram for the Type I and Type II Noninteracting Control Scheme Executive for the Closed Circuit Grinding Simulator

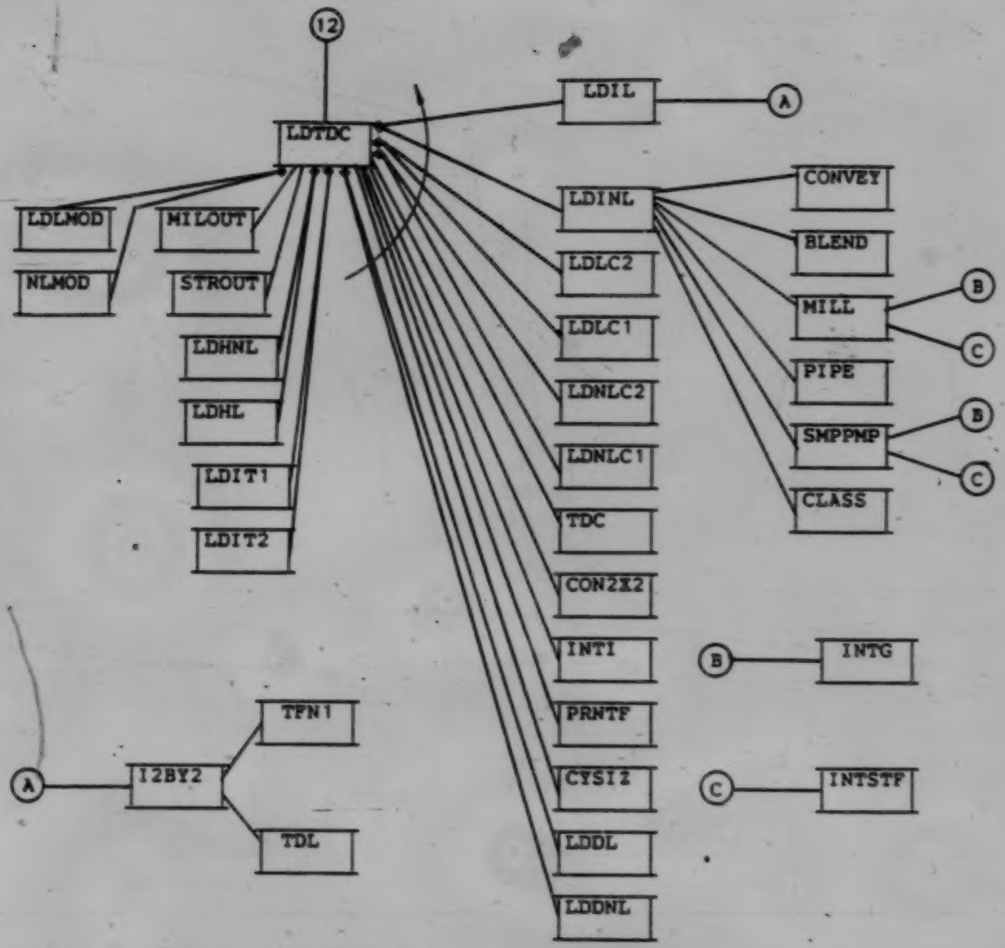

12
LDTDC
LDIL
A
LDLMOD
NLMOD
MILOUT
STROUT
LDHNL
LDHL
LDIT1
LDIT2
LDINL
CONVEY
BLEND
MILL
B
C
PIPE
SMPPMP
B
C
CLASS
LDLC2
LDLC1
LDNLC2
LDNLC1
TDC
CON2X2
INTI
PRNTF
CYSI2
LDDL
LDDNL
B
INTG
C
INTSTF
A
I2BY2
TFN1
TDL

Figure 4.9: Program Structure Diagram for the Type I and Type II Time Delay Compensation Control Scheme Executive for the Closed Circuit Grinding Simulator

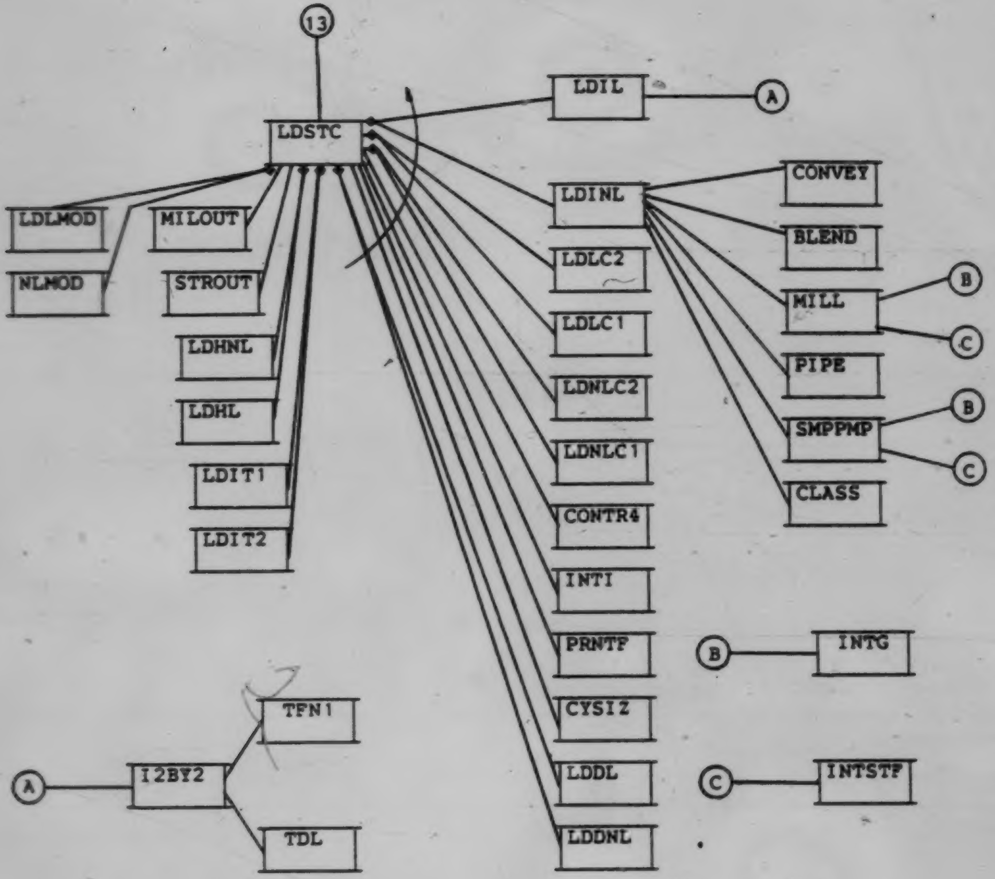

13
LDSTC
LDIL
A
LDLMOD
NLMOD
MILOUT
STROUT
LDHNL
LDHL
LDIT1
LDIT2
LDINL
CONVEY
BLEND
MILL
B
C
PIPE
SMPPMP
B
C
CLASS
LDLC2
LDLC1
LDNLC2
LDNLC1
CONTR4
INTI
PRNTF
CYSIZ
LDDL
LDDNL
B
INTG
C
INTSTF
A
I2BY2
TFN1
TDL

Figure 4.10: Program Structure Diagram for the Type I and Type II Self-Tuning Control Scheme Executive for the Closed Circuit Grinding Simulator

Table 4.6: The Closed Circuit Simulator Subroutines and Functions for the Second Level Executive Programs Specific to Lake Dufault

Subroutine	Function
LDLMOD	This subroutine sets up the linear model for the Lake Dufault grinding circuit.
NLMOD	This subroutine reads the nonlinear model data for the simulator from the data file attached to the logical unit 1 in the run command. The data file for the Lake Dufault grinding circut is named MINSIM.D.
LDHNL	This subroutine prints the PRNTF headings to the simulation output data file attached to the logical unit 6 in the run command if the nonlinear model of the Lake Dufault circuit is used.
LDHL	This subroutine prints the PRNTF headings to the simulation output data file attached to the logical unit 6 in the run command if the linear model of the Lake Dufault circuit is used.
LDIT1	This subroutine initializes the controllers for the Type I configuration of the control system when PI controllers are used with the Lake Dufault circuit.
LDIT2	This subroutine initializes the controllers for the Type II configuration of the control system when PI controllers are used with the Lake Dufault Circuit.
LDDNL	This subroutine is responsible for the open loop disturbance calculations required for the nonlinear model of the Lake Dufault grinding circuit. The ore hardness change function of this subroutine can be used the closed loop executive programs also.

Table 4.6 continued.

LDDL	This subroutine is responsible for the open loop disturbance calculations required for the linear model of the Lake Dufault grinding circuit. The ore hardness change function of this subroutine can be used with the closed loop executive programs also.
LDINL	This subroutine specifies the nonlinear model of the Lake Dufault grinding circuit through CALLs to the appropriate routines in the subroutine library, MIN.LIB.
LDIL	This subroutine specifies the linear model of the Lake Dufault grinding circuit through CALLs to the appropriate routines in the subroutine library, DYFLO2.LIB.
LDNLC1	This subroutine performs the calculations for converting the controller outputs into the manipulated variable values for the Type I configuration and the nonlinear model of the Lake Dufault circuit.
LDNLC2	This subroutine performs the calculations for converting the controller outputs into the manipulated variable values for the Type II configuration and the nonlinear model of the Lake Dufault circuit.
LDLC1	This subroutine performs the calculations for converting the controller outputs into the manipulated variable values for the Type I configuration and the linear model of the Lake Dufault circuit.
LDLC2	This subroutine performs the calculations for converting the controller outputs into the manipulated variable values for the Type II configuration and the linear model of the Lake Dufault circuit.

There are several reasons for developing a second simulator, rather than incorporating its functions into the original closed circuit simulator. The primary reason for this is due the fact that the closed circuit simulator is specific to a particular grinding operation, while the open circuit simulator is somewhat more general. As the two simulators were developed concurrently, some overlap in the interactive functions has arisen, however, because the open circuit simulator does not perform as many functions as the closed circuit simulator, it was not necessary to separate the interactive initialization section from the rest of the executive. Thus, unlike the closed circuit simulator, the executive program developed for the open circuit simulator consists of only a single tier that carries out both the interactive functions along with performing the simulation calculations. It should be noted that the open circuit and the closed circuit simulators could be merged into a single simulator, but at present it is considered to be more logical that the two simulators remain separate, based on the differences in circuit configuration and operation.

The functions performed by the executive program are shown in the data flow diagram in Figure 4.11, and it becomes evident that this program is similar in nature to the second level executives in the closed circuit simulator. The open circuit executive is responsible for interacting with the user to obtain the global simulation data and scheduling the simulation calculations. Figure

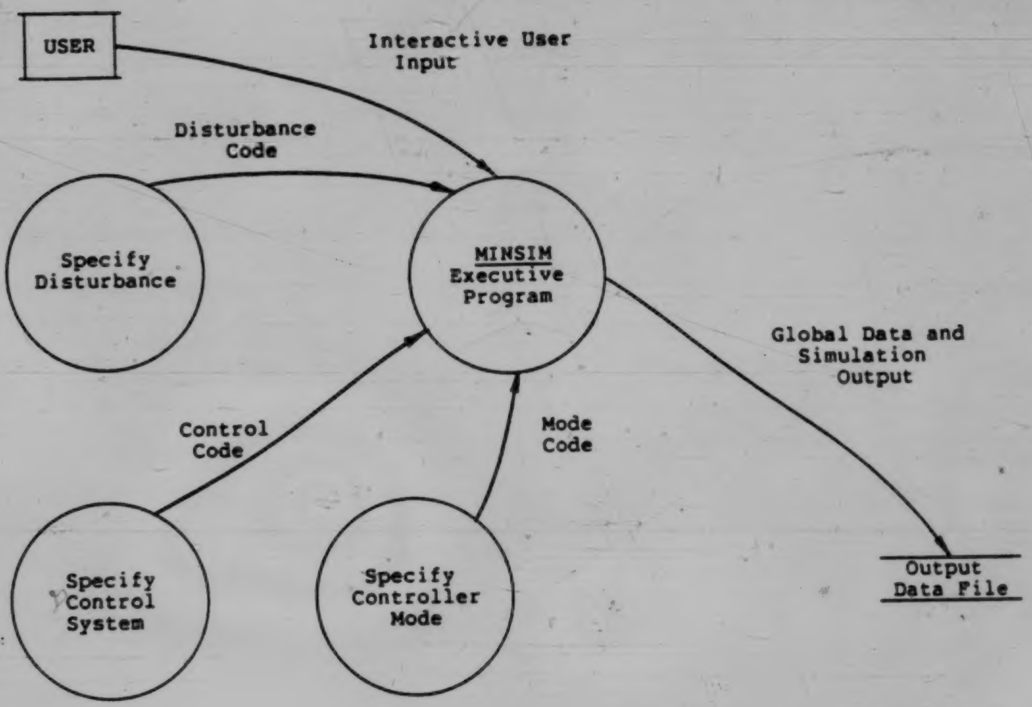

USER
Interactive User Input
Disturbance Code
Specify Disturbance
MINSIM Executive Program
Global Data and Simulation Output
Control Code
Mode Code
Specify Control System
Specify Controller Mode
Output Data File

Figure 4.11: Data Flow Diagram for the Open Circuit Grinding Simulator

4.12 presents a program structure diagram for the simulator where it can be seen that this simulator relies entirely on the subroutine library, DYFLO2.LIB, except for two subroutines used for obtaining user input. It is interesting to note here that the open circuit simulator executive has the same level of complexity as a second level executive of the closed circuit simulator. This however is due only to the fact that far fewer functions were incorporated into the open circuit simulator. Table 4.7 summarizes the the subroutines that are specific to this simulator and gives a brief functional description of each.

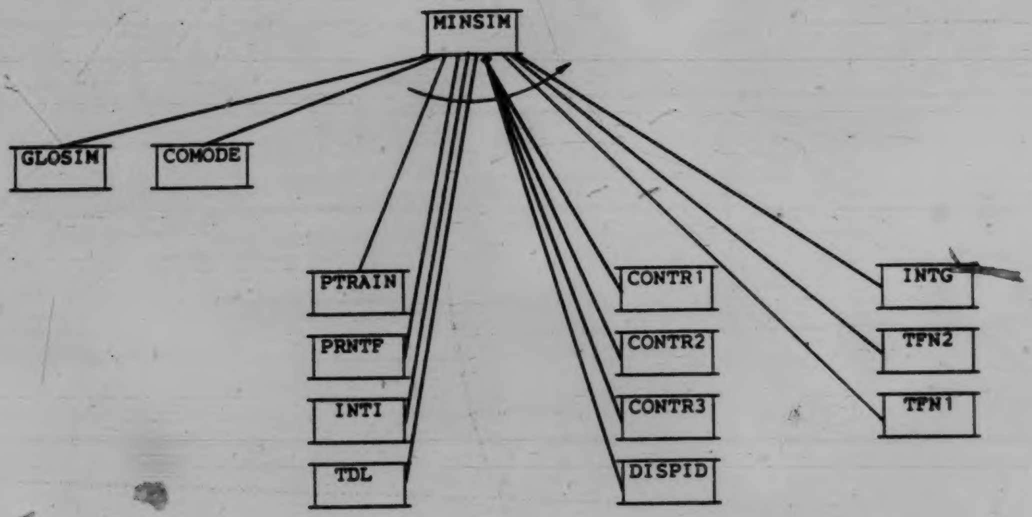

MINSIM
GLOSIM
COMODE
PTRAIN
PRNTF
INTI
TDL
CONTR1
CONTR2
CONTR3
DISPID
INTG
TFN2
TFN1

Figure 4.12: Program Structure Diagram for the Executive of the Open Circuit Grinding Simulator

4.4 User Interface Description

The MINSIM software package, as mentioned earlier, is a menu driven system of subroutines linked through executive programs. The system is reasonably user friendly and will trap most fatal errors before crashing the program. Although prompting which includes typical user input has been provided when appropriate, a certain level of user sophistication has been assumed with respect to process control theory. It is also expected that the user is somewhat familiar with the particular circuit before attempting to use the simulator. This section provides all the information required for running the MINSIM simulation

Table 4.7: The Open Circuit Grinding Simulator Subroutines and Functions

Subroutine	Function
MINSIM	The executive program. This program schedules the interactive user input to the simulation run and calls the appropriate subroutines from DYFLO2.LIB to perform the simulation calculations.
COMODE	This subroutine allows the user to specify the controller mode (P, PI, or PID) and the appropriate settings.
GLOSIM	This subroutine allows the user to specify global simulation parameters such as the integration interval and total simulation time.

package at the University of Alberta and gives a brief overview of the various options available. The reader is encouraged to read Chapters 5 and 6 before using the system to gain additional insight into its many capabilities.

4.4.1 The Closed Circuit Grinding Simulator

The circuit used to illustrate the features of the closed circuit grinding simulator is the Lake Dufault grinding circuit (Smith and Guerin, 1980), either in an open loop configuration or under an number of different multi-input multi-output control schemes. Instructions are available online and the user can choose between a demonstration run, where a set of predetermined simulation parameters are used, and a user specified run, where all simulation parameters are set by the user. This enables the new user to understand various features of MINSIM without a detailed knowledge of its functions while the more experienced user is able to experiment with various controller settings and disturbances. For either type of run, the particular second level executive to be used is specified when the control system is chosen and the disturbance input is set by the user. The advanced user is allowed to choose the magnitude of the disturbance as well as the time it enters the system.

Although there are no specific equipment requirements for MINSIM, it is suggested that a CRT type terminal be used. Before running the simulator, some initial preparation is required. The user must have READ access to the data files MINSIM.D, STC.D, and PULSE.D, as well as the object code versions of the subroutine libraries DYFLO2.LIB, MIN.LIB, and STC.LIB, on the CSid account, GMCD, of the MTS at the University of Alberta. READ access to the object code version of the first and second level executive programs which comprise the heart of the closed circuit simulator are also required and reside in the file MINSIM.2.7 on the same account. The execution macro required to run this version of MINSIM is in the file MINSIM.2.EXE. It should be noted that the FORTRAN source code listings of these subroutine libraries are available on the account, GMCD, in the files S.DYFLO2.LIB, S.MIN.LIB and S.STC.LIB while the source code for the executive programs is in the file S.MINSIM.2.7.

Before the simulation session can begin, the user must first issue the MTS command:

```
$SOURCE GMCD:MINSIM.2.EXE
```

to initialize the simulator execution macro, MINSIM2. To run the simulator, the user simply issues the command:

```
$MINSIM2
```

This macro initializes the simulation output data files and then issues the MTS command:

```
$RUN GMCD:MINSIM.2.7+GMCD:DYFLO2.LIB+GMCD:MIN.LIB+
GMCD:STC.LIB 1=MINSIM.D 2=STC.D 3=PULSE.D 6=-MIN.OUT
7=*SINK* 8=-STC.OUT 9=-STC.PAR1 10=-STC.PAR2
11=-STC.SET  T=15
```

which links the appropriate subroutine libraries to the simulator executive and assigns the input/output logical units to the proper files. It should be noted that the local CPU time estimate in the above command is set to an extremely large value because this much time is usually required to run the self-tuning controller option of the simulator.

A sample terminal session for this simulator can be found in Appendix G. The user is first greeted with the simulator title and version number, and then asks if instructions are necessary. If the user types "Y" and the return key ("ENTER", "RETURN", etc.), the simulator will provide some brief instructions on the functions of the simulator including some warnings about the cost. The first menu will appear next and the user is required to choose between the demonstration and user specified modes of operation. An integer number should be entered here and the demonstration mode is suggested for first time users because it sets up much of the simulation automatically. Again, the user must type the return key to continue. The next menu in both simulation modes is the simulation circuit menu. At this point the user must choose the control system configuration to use in the simulation. Nine options are available including open loop, classical Type I or Type II control, mulitloop (decoupling) control, time delay compensation control, and self-tuning control. An invalid response will be rejected and the user is prompted to reenter the desired choice until a valid option is chosen. It should be noted that the self-tuning option is not available in the demonstration mode.

Following this menu, if the user has chosen the demonstration mode, the simulator will output a summary of the parameters being used in the simulation and a prompt will be issued asking if the information is correct. Entering "N" will cause the simulator to cycle back to the simulation type menu, while "Y" will complete the execution of the simulation run. The simulation output file is a temporary file called -MIN.OUT, and can be copied to the terminal screen or to a printer. It should be noted that the demonstration mode simulates the operation of the Lake Dufault grinding operation for 100 minutes using the linear model of the circuit.

If the user specified mode was chosen from the simulation type menu, the user is prompted for the model type (linear or nonlinear) from the model type menu. If an invalid option is entered it will be rejected and the user prompted to reenter the desired choice until a valid option is chosen. At this stage, the user must specify the integration interval, the total simulation time, and the print interval for the output data. This data must be entered as real numbers (i.e. use a decimal point) or the program output will be erroneous. The user must next provide various controller constants, including the gain, integral time, and setpoint for each loop if any option but number 1 (open loop) is used. If either of the multiloop options are chosen, decoupling controller constants, such as the gain, the lead time constant, and the lag time constant, must also be specified. The expected response format is given in brackets following the prompt using typical values for the Type I control configuration. Again, the input format expects real numbers.

The program branches here depending on the circuit configuration option. The open loop circuit option (number 1) causes the open loop disturbance menu to appear, while options 2 through 7 cause the closed loop disturbance menu to appear. The disturbances available for the open loop simulation are step changes in the fresh ore feed (FOF) flow rate, the sump water feed (SWF) flow rate, and the fresh ore hardness (HRD). The closed loop simulation allows step changes in the hydrocyclone overflow size (COS) specification setpoint (loop 1), the sump pulp level (SPL) setpoint (loop 2), as well as changes in ore hardness. In either case, the user must enter the integer number corresponding to the desired disturbance. The magnitude of the disturbance and the time it is to enter the control system must also be specified. A real number format must be used for these last two inputs.

A third branch is taken if either of the self-tuning options are chosen. These options require the user to choose between disturbances to either the loop 1 or loop 2 setpoints for parameter identification, and then the duration of the pulse train setpoint forcing function. It is suggested that the duration be set to at least ten minutes. The fresh ore hardness disturbance is always used by the self-tuning controller option and the magnitude and time to enter the system must next be specified. In this case, the disturbance time is specified in terms of the pulses duration. The suggested input is a real number greater than five because the pulse train specified in the data file, PULSE.D, contains five steps. The sample interval for both loops is requested next. The data file, STC.D, is set up to work with a sample interval of 0.3 minutes for both loops and it is suggested that this not be shortened due to computing cost considerations. Additional self-tuning controller initialization data such as when identification and self-tuning control is to start and how often the self tuning controller statistics are to be printed is requested at this point. A summary of all the required input for the self tuning option is provided in Table 4.8 along with suggested responses.

The final step before the simulation calculations are performed is to print a summary of the simulation parameters to the terminal/screen and prompt the user to check if the values are correct. Entering "Y" causes the simulation to proceed, while the simulator will recycle to the simulation type menu if "N" is entered. The user is presented with a list of names of temporary files where the simulation output may be found when the simulation is complete. Table 4.9 presents a summary of the contents of the various temporary output files created by the program, and Table 4.10 gives a summary of all the options available in the simulator. It should be noted that subsequent runs during a given terminal session can be accomplished by entering the command:

```
$MINSIM2
```

as described earlier.

The sample terminal session presented in Appendix G is for the user specified option with a classical Type I control strategy. This material should provide the needed background to appreciate use of the Lake Dufault version of the MINSIM software package. The simulation output file, -MIN.OUT, for this run is also given in Appendix G.

Table 4.8: Self-Tuning Control Option Interactive Input Summary

Input Description	Suggested Response	
	Loop 1	Loop 2
Identification setpoint step input disturbance loop number. Either loop may be chosen, however, loop 1 (COS) is suggested.	1 or 2	1 or 2
Setpoint pulse disturbance duration in minutes. Ten minutes is suggested.	10.	-
Absolute value of the setpoint pulse magnitude in per cent.	1.5	-
Hardness disturbance entry time in terms of the number of pulse duration intervals.	10.	-
Hardness disturbance magnitude in per cent.	50.	-
Discrete control sample interval in minutes.	0.3	0.3
Identification start time in terms of sample intervals. It is suggested that parameter identification begin immediately.	0	0
Self-tuning controller start time in terms of sample intervals. This actuates the self-tuning controller action after the initial identification stage.	117	117

Table 4.8 continued.

Self-tuning controller data output interval in terms of sample intervals. This prints various self-tuning controller statistics and the covariance matrix every 20 minutes of simulation time to the file, -STC.OUT .	67	67
Disturbance measurement type. The ore hardness disturbance is not measurable online.	0	0
Covariance matrix print option. The printout is optional (Y/N) with cost reduced slightly by not printing.	Y/N	Y/N

4.4.2 The Open Circuit Grinding Simulator

This simulator is capable of simulating the response of an open circuit grinding operation either for determining open loop dynamics or under control using continuous or discrete P, PI, or PID controllers in single input-single output feedback control schemes. The grinding circuit model is based on work by Adel et al. (1983) and this simulator is intended to reproduce the results obtained by these authors.

The execution procedure for this version of MINSIM is much like that described for the open circuit simulator. The user must first ensure that there is READ access to the subroutine library file, DYFLO2.LIB, and the executive program file, MINSIM.3.1, on the CSid GMCD. The execution macro for this simulator is in the file MINSIM.3.EXE on the same account. It should be noted that the FORTRAN source code listing for the executive program

Table 4.9: Closed Circuit Simulator Temporary Output File Summary

File Name	Description of Contents
-MIN.OUT	This file contains the global simulation input data including the circuit, model, integration interval, total simulation time and controller constants along with the simulation output data (using PRNTF). This file is the main output file for all circuit options.
-STC.OUT	This file contains self-tuning controller statistics and the covariance matrix output and is used only when the self-tuning controller option is specified.
-STC.PAR1	This file contains the parameter estimates for loop 1 when the self-tuning controller option is specified.
-STC.PAR2	This file contains the parameter estimates for loop 2 when the self-tuning controller option is specified.
-STC.SET	This file contains the setpoints for both control loops when the self-tuning controller option is specified.

file for this simulator is in the file S.MINSIM.3.1 on the account, GMCD.

Before the simulation session can begin, the user must first issue the command:

```
$SOURCE GMCD:MINSIM.3.EXE
```

to initialize the simulator execution macro, MINSIM3. To run the simulator, the user simply issues the MTS command:

```
$MINSIM3
```

Table 4.10: Closed Circuit Grinding Simulator Menu and Option Summary

Menu Title	Option Description
Simulation Type	1 - Demonstration: Simulator demonstration mode, suggested for first time users. 2 - User Specified: Advanced user mode, allows experienced users to alter many simulation parameters.
Simulation Circuit	1 - Open loop simulation. 2 - Classical Type I control. 3 - Classical Type II control. 4 - Noninteracting Type I control. 5 - Noninteracting Type II control. 6 - Time delay compensated Type I control. 7 - Time delay compensated Type II control. 8 - Self-tuning Type I control. Note: This option does not function in demonstration mode. 9 - Self tuning Type II control. Note: This option does not function in demonstration mode.
Model Type	1 - Nonlinear model. 2 - Linear model. Note: This menu does not function in demonstration mode.

Table 4.10 continued.

Open Loop Disturbance Type	1 - Fresh Ore Feed (FOF) flow rate step input change. 2 - Sump Water Feed (SWF) flow rate step input change. 3 - Hardness (HRD) step change. Note: This menu does not function in demonstration mode.
Closed Loop Disturbance Type	1 - Loop 1 (COS) setpoint step change. 2 - Loop 2 (SPL) setpoint step change. 3 - Hardness (HRD) step change. Note: This menu does not function in demonstration mode.
Self-Tuning Controller Loop Disturbance	1 - Loop 1 (COS) setpoint pulse. 2 - Loop 2 (SPL) setpoint pulse. Note: This menu does not function in demonstration mode.

This macro initializes the simulation output data file and then issues the command:

```
$RUN GMCD:MINSIM.3.1+GMCD:DYFLO2.LIB 6=-MIN.OUT
7=*SINK* T=10
```

which links the appropriate subroutine libraries to the simulator executive and assigns the input/output logical units to the proper files. It should be noted that the CPU time estimate in the above command is set to a large value because the pulse option of the simulator usually needs six seconds to execute.

A sample terminal session for this simulator can be found in Appendix G. The user is first greeted with the simulator title and version number, and then asked if the user wishes to continue. If the user types "Y" and the return key, the program issues a prompt for the integration interval. Following this, the simulation time and output data print interval must be specified. It should be noted here that the format code for these inputs is real. Unpredictable results will occur if real values are not entered.

The control system menu appears next, and the user is prompted to enter the integer code specifying the type of control system to be used for the simulation. An invalid option will be rejected and the user is prompted to reenter a valid integer. The next menu allows the user to choose the controller mode, and then requests the appropriate controller settings as real numbers.

The final menu to be exhibited before the simulation run is completed is the disturbance menu. The user must enter an integer number corresponding to the disturbance code and then specify time in terms of hours at which it is to enter the control system. The particle size analysis time delay (hours) is requested here also. After the simulation calculations are complete, the user is presented with the temporary file name, -MIN.OUT, where the simulation output may be found. Table 4.11 provides a summary of all the options available in the simulator. It should be noted that subsequent runs during a given terminal session can be accomplished by reissuing the command:

```
$MINSIM3
```

described earlier.

The sample terminal session presented in Appendix G utilizes a continuous PI controller with the grindability/surface area step disturbance and should provide some additional insight into the use of the open circuit version of the MINSIM software package. The simulation output file, -MIN.OUT, for this run is also given in Appendix G.

4.5 Update Procedure

A modular subroutine programming approach has been taken in the implementation of all the software in the MINSIM simulation package. Obvious reasons for this programming method are:

(i) Ease of software maintenance. The logic tends to be simpler to understand and update bacause all the code associated with a specific function is grouped together. Typically, a subroutine module consists of ten to fifteen lines of executable code.

(ii) Each module is easily verified. A simple executive program can usually be written to test the results produced by new subroutines against published data.

(iii) Computer memory is used more efficiently. Generalized subroutines are used rather than duplicate coding.

Table 4.11: Open Circuit Grinding Simulator Menu and Option Summary

Menu Title	Option Description
Control System	1 - Open loop simulation. 2 - Continuous control simulation. 3 - Discrete control simulation.
Controller Mode	1 - Proportional only. 2 - Proportional plus integral (PI). 3 - Proportional plus integral plus derivative (PID).
Disturbance	1 - Grindability/surface area step input. 2 - Grindability/surface area pulse train input.

(iv) Compatability with the existing DYFLO2 software is maintained.

Development of new subroutine modules to increase the functional capabilities of the three subroutine libraries discussed in Section 4.2 should be written with the above four points in mind. By utilizing the existing COMMON blocks and as much of the existing software as possible, programming effort can be kept to a minimum and functional duplication avoided.

4.5.1 Subroutine Library Update

When a subroutine module is completely tested and ready to be included in a subroutine library, the procedure listed below can be used to place it in the appropriate library.

1. From MTS, issue the commands:

   ```
   $RUN *FORTG SCARDS=subroutine source code file name
   ```

   This compiles the subroutine under the FORTRAN IV compiler, placing the object code in the temporary file, -LOAD#.

2. `$RUN *OBJUTIL 0=subroutine object code library name`

   This command runs the object utility system program which enables the programmer to manipulate object code libraries. It is strongly suggested that the programmer unfamiliar with this facility read the Computing Services documentation before attempting to use this program.

3. Next, issue the *OBJUTIL command:

   ```
   ADD -LOAD#
   ```

   This will add the subroutine object code to the subroutine library attached to the logical unit, 0, in the $RUN command in step 2. *OBJUTIL will respond with:

   ```
   subroutine name ADDED
   ```

4. Finally, to exit from *OBJUTIL, the programmer must issue the command:

   ```
   MTS
   ```

   to return to the MTS command mode.

4.5.2 Simulator Executive Update

The procedure for adding new features to either of the simulator executive structures described in Section 4.3 is somewhat more involved than adding a new function to the subroutine libraries. This is because the programmer must provide for interactive user input and change the menus to reflect the new functions available in addition to the code that accomplishes the new function. A good understanding of the DYFLO2 executive programming technique is essential for successfully completing any update. The following discussion assumes that the programmer has a good working knowledge of both FORTRAN and DYFLO2.

Of the two simulators, the open circuit version requires the most advanced knowledge of the MINSIM software package due to the number and complexity of its functions. Adding a new control scheme to this simulator is the only update that will be considered in this discussion. The suggested procedure is as follows:

1. Copy the second level executive program that most closely performs the functions required of the new function to a separate file. This provides a standard framework on which the new second level executive can be based and will ensure consistency between the various executive programs.
2. Delete the initial comment lines and SUBROUTINE definition line. At this time any unnecessary code can be deleted as well.
3. Alter the initialization section of the program to

include the initialization data normally provided interactively by the user. It is suggested that this be kept together so that it may be easily deleted when the new executive is ready for incorporation into the simulator.

4. Add the new code required to perform the new feature, including comment statements describing the function. Most of the changes should likely appear in the derivative and integration sections of the program, as these sections perform the recursive calculations for controlling the grinding circuit. Debug this program by compiling and running it separately from the rest of the simulator. It should be noted that it will be necessary to link the compiled version of the original simulator to this program in order to gain access to the controller and model intitialization subroutines. It should be noted that several logical units are potentially referenced by a given second level executive program. Table 4.12 provides a list of all those currently assigned.

5. Once the new executive program functions correctly, it can be incorporated into the simulator by deleting the initialization data and inserting a SUBROUTINE declaration statement into the program with a name indicating its function. The subroutine parameter list must include the model type and input output variable pairing configuration flags. As well, access to any special user specified input data must be provided through the user interface, or in an extreme case, through a new COMMON block.

Table 4.12: Logical Units Referenced by the Closed Circuit Simulator

Logical Unit Number	Attached to file	Description of Contents
1	MINSIM.D	Initialization data for the Lake Dufault grinding circuit nonlinear model.
2	STC.D	Self-tuning controller initialization data for the Lake Dufault grinding circuit.
3	PULSE.D	Pulse train definition file for the setpoint pulse input used for the self-tuning control configurations of the Lake Dufault simulator.
6	-MIN.OUT	See Table 4.9.
7	*SINK*	Terminal I/O.
8	-STC.OUT	See Table 4.9.
9	-STC.PAR1	See Table 4.9.
10	-STC.PAR2	See Table 4.9.
11	-STC.SET	See Table 4.9.

6. The next step is to modify the interactive subroutines in the original simulator. The new function must then be added to the simulation circuit input subroutine, SIMCIR, as an option in the simulation circuit menu. The input error checking in this subroutine must be changed to reflect the new option as well. If the new option is to run in the demonstration mode, appropriate initialization data must be added to the demonstration initialization subroutine, DSIMIN. It is suggested that if the demonstration mode is not implemented for the new option that provisions be made to inform the user in the demonstration initialization subroutine as has been done for the self-tuning controller option. Other special user input should be incorporated into the interactive simulation input subroutine, ISIMIN.

7. The global simulation parameter output subroutine, SIMOUT, must be updated to reflect the new option. This involves setting up new FORMAT statements to write the appropriate information to the simulation output file attached to logical unit 6.

8. Finally, the first level executive program must be altered to include a CALL to the new second level executive subroutine.

It is not expected that the open circuit grinding simulator will require updating due to its specialized nature.

5. Lake Dufault Grinding Circuit Simulation Study

5.1 Introduction

With the closed circuit simulator described in the previous chapter, various different grinding circuit flowsheets under different control schemes can be simulated. The purpose of this chapter is to present the results of a simulation study of an existing circuit, that of the Lake Dufault division of Falconbridge Copper Mines Limited.

In the past, it was typical to control grinding circuits manually, using "grab" samples as an indication of grinding performance. Later, vibration monitors were located near the mill casing to measure the sound production and combined with small process computers to detect overload conditions. More recently, the mineral processing industry has been able to benefit from the use of controllers and control techniques~~, developed~~ for the chemical industry. The objective of this chapter is to demonstrate the usefulness of a dynamic grinding circuit simulator and to explore the advantages associated with different control system configurations.

The following section begins with a description of the actual circuit and outlines the appropriate parameters associated with the models presented in Chapter 3. Following this, open loop responses to various disturbances are discussed in Section 5.3 and a 2x2 transfer function model of the plant is established in Section 5.4. A method for estimating the initial settings for proportional plus integral (PI) controllers is given in Section 5.5. Section 5.6 presents results for two multiloop control schemes typically used on such circuits in industry. Time delay compensation is the first, more advanced control technique to be applied to the original plant. This controller configuration is discussed in Sections 5.7 and 5.8. Multivariable control techniques are next applied to the plant. An interaction analysis is performed and a dynamic decoupler designed in Section 5.9 and these results (through simulation) are described in Section 5.10. This chapter concludes with a discussion on adaptive control in Sections 5.11 and 5.12.

5.2 Grinding Circuit Description

The flowsheet of the Lake Dufault grinding circuit is shown in Figure 5.1. Fresh ore is fed onto a variable speed conveyor from the crushed ore bins and then onto a fixed speed conveyor where it is weighed. Fresh water is added and the slurry proceeds into the rod mill for grinding. The ground product overflows from the rod mill and is gravity fed to the sump. Additional water is added at this stage and the mixture is pumped into the hydrocyclone for classification. A density gauge is provided between the pump and the hydrocyclone for monitoring the cyclone feed density (CFD). In the hydrocyclone, the slurry is

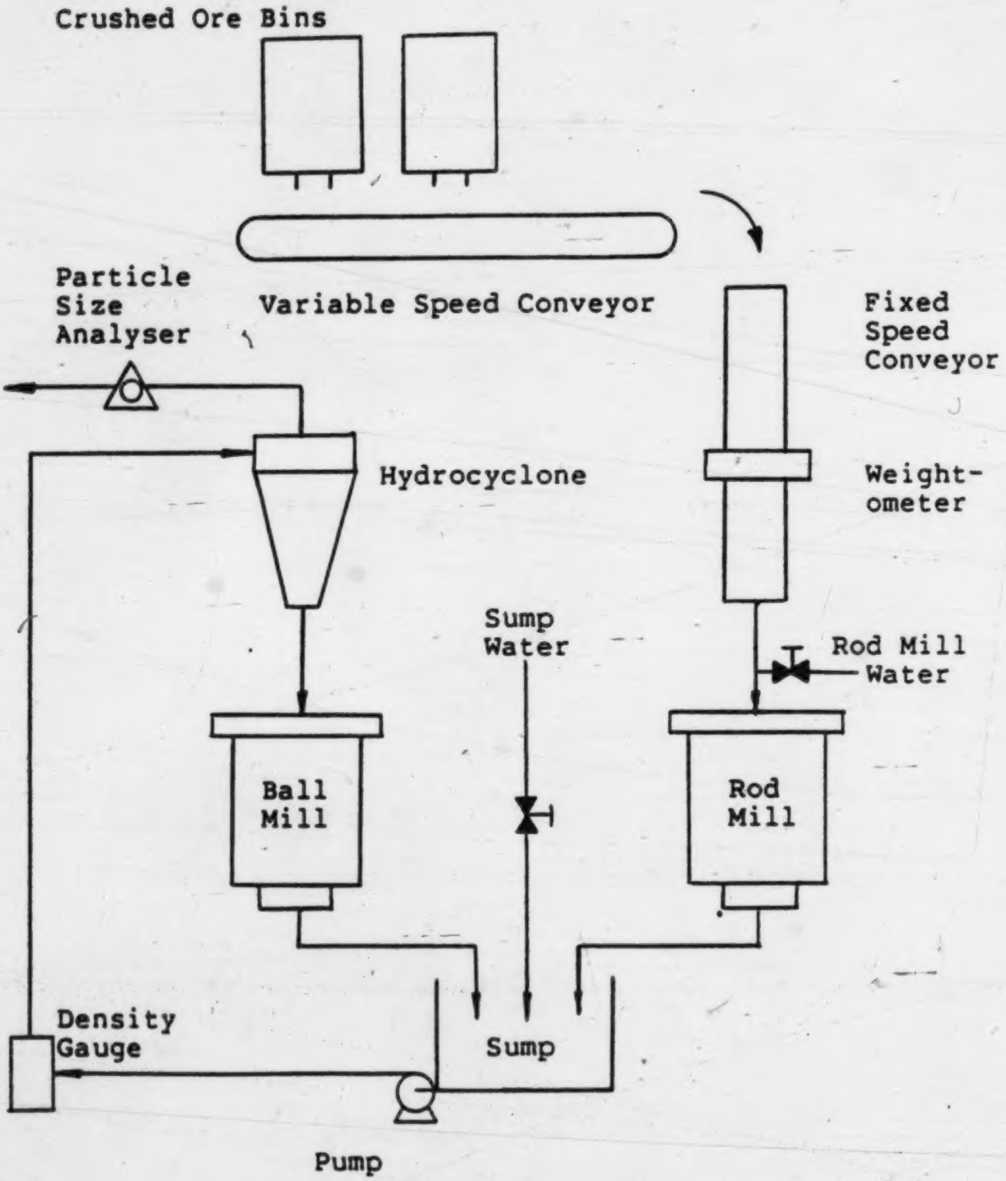

Crushed Ore Bins
Particle Size Analyser
Variable Speed Conveyor
Fixed Speed Conveyor
Hydrocyclone
Weight-ometer
Sump Water
Rod Mill Water
Ball Mill
Rod Mill
Density Gauge
Sump
Pump

Figure 5.1: Schematic Diagram of the Lake Dufault Closed Circuit Grinding Operation

classified by particle size. The undersize slurry stream exits through the vortex finder (hydrocyclone overflow) to a particle size monitor and then exits as the final product. The oversize particles (hydrocyclone underflow) are fed into a ball mill for further grinding. The ball mill product is recirculated to the sump, making this a closed circuit grinding operation.

Typically, particle size monitors are subject to frequent breakdowns due to plugging and are considered to be highly unreliable devices. This problem is overcome by utilizing the density gauge readings for the hydrocyclone feed stream along with the inlet pressure and flowrate of solids and water (Seitz and Kawatra, 1984) to infer the overflow particle size distribution. This is facilitated by implementing a hydrocyclone classifier model such as that presented in Chapter 3 (cf. Section 3.3) on a process control computer. For the duration of this work, it will be assumed that such a situation exists and that the hydrocyclone overflow size distribution is accurately represented.

Typical controlled variables in such a circuit include the hydrocyclone feed density (CFD), the hydrocyclone overflow particle size distribution (COS), and the sump pulp level (SPL). The manipulated variables available to control these variables are the fresh ore feed rate (FOF), the rod mill fresh water addition rate and the sump water feed rate (SWF). The primary disturbance to this type of circuit is the fresh ore feed hardness (HRD) and usually goes unmeasured. Other possible disturbances include feed particle size distribution changes, water flow rate disturbances and sudden changes in mill behavior. These latter disturbances are not considered important in this work and will not be discussed further.

Each of the pieces of equipment shown in Figure 5.1 can be modelled as a separate unit operation or as a time delay. All the required models were discussed in Chapter 3 and Figure 5.2 presents a schematic representation of the simulation flow sheet. Stream numbers are included on this figure for convenience, and correspond to those used in the simulator. The fresh ore feed enters as stream 1 where it is delayed by a fixed time delay (FTD) representing the fixed speed conveyor in Figure 5.1. The exit stream from the conveyor is blended with the rod mill feed water in a blender (B) and then fed to the rod mill. The rod mill consists of a call to the mill subroutine (M) in series with a call to the variable time delay (VTD) routine to model the plug flow element of the transport behavior normally associated with tumbling mills. The output stream from the VTD is blended with the sump water feed (stream 7) and the output from the ball mill (stream 14). The sump and pump model routine (SAP) is called, and the pump output stream is fed to the hydrocyclone (HC) routine. The underflow (stream 12) particle size distribution is calculated along with the water and mass flow rates and this information is passed to

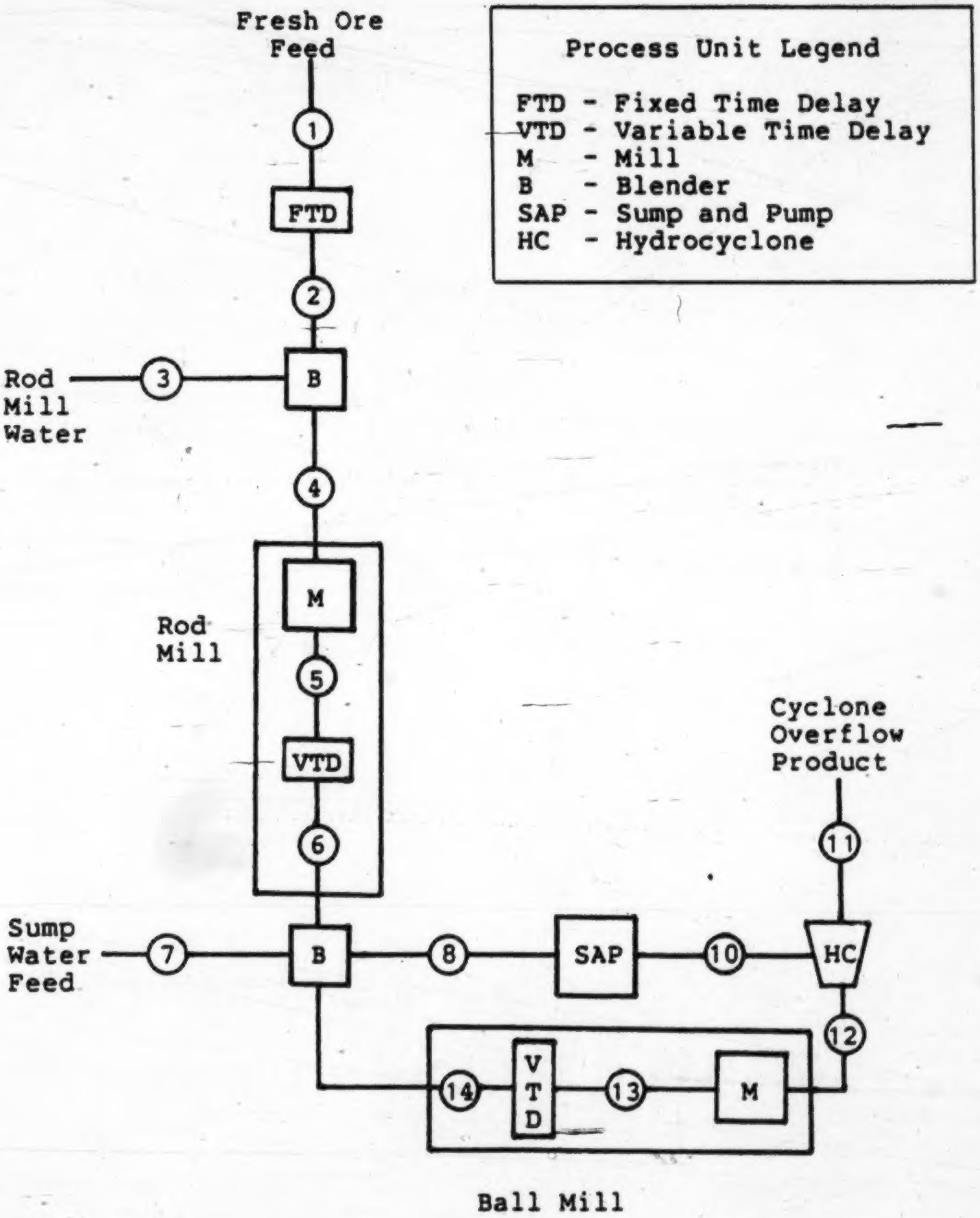

Fresh Ore Feed
Process Unit Legend
FTD - Fixed Time Delay
VTD - Variable Time Delay
M - Mill
B - Blender
SAP - Sump and Pump
HC - Hydrocyclone
1
FTD
2
Rod Mill Water
3
B
4
Rod Mill
M
5
VTD
6
Cyclone Overflow Product
11
Sump Water Feed
7
B
8
SAP
10
HC
12
14
V T D
13
M
Ball Mill

Figure 5.2: Simulation Flowsheet for the Lake Dufault Grinding Circuit

the mill model for the ball mill calculations. The hydrocyclone overflow (stream 11) product mass flow rates are also calculated by the hydrocyclone routine from a material balance around the hydrocyclone. The mill routine is again cascaded with the variable time delay routine to complete the ball mill calculations.

Before using the simulator, it is necessary to complete the models by defining the parameters for the Lake Dufault circuit. The FTD has arbitrarily been assigned a value of 0.15 minutes. Each grinding mill is modelled using equations of the form given in equations (3.18) and (3.19). The first parameter that must be set is the number of screen sizes, n, to be used in defining the particle size distribution. A larger number, in general, produces a model that is more capable of providing accurate results. Since computational effort increases rapidly as the number of size classes are increased, a practical limit of 14 to 16 is usual. For this work, 16 particle size classes were chosen. The breakage function, b, was given by equation (3.24) for both types of mill depending on the choice of the parameter, a. With $n=16$, the first column of the breakage matrix for each mill is calculated and summarized in Table 5.1.

Using equations (3.25) and (3.26), the selection functions for the two mills in the Lake Dufault circuit are calculated and summarized in Table 5.2.

The initial steady state feed and product particle size

Table 5.1: First Column of the Breakage Matrix for the Lake Dufault Grinding Circuit

Lower Bound on Particle Size (microns)	Rod Mill	Ball Mill
4757	0	0
3364	.209	.202
2378	.165	.161
1682	.131	.129
1189	.103	.103
841	.082	.082
595	.065	.065
420	.051	.052
297	.041	.042
210	.032	.033
149	.025	.027
105	.020	.021
74	.016	.017
53	.013	.013
37	.010	.011
10	-	-

Table 5.2: First Order Selection Constants for the Lake Dufault Grinding Circuit

Lower Bound on Particle Size (microns)	Rod Mill (minutes^{-1})	Ball Mill (minutes^{-1})
4757	22.950	17.804
3364	4.860	9.483
2378	1.028	5.047
1682	0.218	2.690
1189	0.046	1.432
841	0.021	0.763
595	0.021	0.407
420	0.021	0.216
297	0.021	0.115
210	0.021	0.061
149	0.021	0.033
105	0.021	0.017
74	0.021	0.009
53	0.021	0.005
37	0.021	0.003
10	-	-

distributions for the rod and ball mills are presented in Table 5.3 in terms of weight percent retained in the size classification. These data were derived from size population data given by Smith and Guerin (1980) and then running the simulator without any disturbances. The size distributions given in Table 5.3 are the steady state values that resulted and of course, the product streams are taken as representative of the internal properties of both mills. It should be noted that one more size classification was used in describing the ball mill particle size distribution than that indicated by Smith and Guerin (1980) to maintain consistency in the population balances throughout the circuit.

The residence time of each mill varies with the inlet volumetric flow rate, Q (cf. equation (3.17)). In order to calculate the residence time, the mill volumetric holdup, V, is needed. For the rod and ball mills used in the Lake Dufault circuit, the effective volumetric holdup (after exclusion of rods and balls) is 3.9 m^3 and 15.9 m^3 respectively. Based on initial steady state volumetric flow rates of 38.8 m^3/hr for the rod mill and 82.2 m^3/hr for the ball mill, the nominal residence times are 0.1 and 0.2 hours, respectively. The variable time delay elements in both mills are arbitrarily assigned nominal values of one minute. The number of blocks used in the variable time delay model for the rod mill is four, while for the ball mill five blocks are used.

Table 5.3: Steady State Particle Size Frequency Distribution for the Lake Dufault Rod and Ball Mill Feed and Product Streams

Lower Bound on Particle Size (microns)	Rod Mill (wt %)		Ball Mill (wt %)	
	f_i	p_i	f_i	p_i
4757	61.00	0.44	0.24	0.0
3364	8.00	0.68	0.37	0.0
2378	6.00	2.80	1.54	0.03
1682	5.24	8.68	4.85	0.16
1189	2.84	13.27	7.66	0.51
841	1.80	12.19	7.57	1.05
595	1.60	10.10	7.20	1.98
420	1.04	8.01	7.22	3.54
297	1.09	6.80	8.18	6.05
210	1.16	5.79	9.18	9.09
149	1.21	4.86	9.34	11.39
105	1.17	4.23	8.30	11.87
74	1.13	3.64	6.66	10.77
53	1.36	3.40	5.23	9.13
37	0.52	2.27	3.66	7.08
10	4.84	12.75	12.80	27.34

The various constants for the hydrocyclone model have been provided by Smith and Guerin (1980) with the constants determined from actual hydrocyclone operating data to be:

$K_1 = 32.42$

$K_2 = -9.13$

$K_3 = 2.87$

$K_4 = 0.0145$

$K_5 = -0.0202$

The separation sharpness parameter, γ, was found to be 0.89. It should be noted that one significant difference between the model presented for the hydrocyclone in Chapter 3 and the model used in simulating the Lake Dufault circuit exists. Equation (3.29), which defines the water split, was found to have a slope of 0.752, rather than 1.1. In other words, equation (3.29) for this circuit is:

$$W_o = 0.752W_f + K_2 \qquad (5.1)$$

Smith and Guerin (1980) offer the fact that the circuit was operating at 140% of its design capacity as an explanation for this difference.

The steady state inlet and outlet particle size distributions for the hydrocyclone were determined in a similar fashion to that described earlier for the mills. Data given by Smith and Guerin (1980) was used as the initial starting point and the simulator was allowed to run to steady state without any disturbances. Table 5.4 summarizes these results. Of course, the underflow particle size distribution matches the ball mill feed stream particle size distribution because these are the same stream. It should be noted once again that one more particle size classification was used than that indicated by Smith and Guerin (1980) to maintain consistency in the population balances throughout the circuit. The product size specification used in this work is based on the percentage of solid material in the hydrocyclone overflow stream that passes through a 100 mesh screen (i.e. smaller that 150 microns). Using the particle size distribution for the overflow stream given in Table 5.4, 81.048 mass percent of the product stream passes 100 mesh at steady state.

The lift height from the pump centre line to the hydrocyclone inlet is assumed to be 12.2 metres and the

Table 5.4: Steady State Particle Size Frequency Distribution of the Lake Dufault Hydrocyclone Feed and Product Streams

Lower Bound on Particle Size (microns)	Feed (wt %)	Overflow (wt %)	Underflow (wt %)
4757	0.15	0.0	0.24
3364	0.24	0.0	0.37
2378	1.00	0.0	1.54
1682	3.15	0.0	4.85
1189	4.98	0.01	7.66
841	4.96	0.11	7.57
595	4.83	0.42	7.20
420	5.11	1.18	7.22
297	6.31	2.85	8.18
210	7.94	5.62	9.18
149	9.14	8.76	9.34
105	9.20	10.86	8.30
74	8.27	11.24	6.66
53	7.12	10.62	5.23
37	5.40	8.62	3.66
10	22.23	39.71	12.80

diameter of the pipe used to connect the hydrocyclone to the pump is 20.3 centimetres (8.0 inches) (Wong, 1984). The density of the hydrocyclone feed pulp is 1.65 tonnes/m^3 and the nominal operating pressure of the hydrocyclone is 36.5 kPa (Smith and Guerin, 1980). The pump used in this circuit is a 25.4 cm x 20.3 cm (10 in. x 8 in.) S.R.L. pump operating at 700 RPM (Flintoff et al., 1985). The total dynamic head (TDH) for this pump is described by equation (3.41) with the following constants:

$K_{P1} = 21.934$

$K_{P2} = -3.414 \times 10^{-3}$

$K_{P3} = -1.454 \times 10^{-5}$

The steady state pulp volumetric flow rate to the hydrocyclone is 168.8 m^3/hour which is slightly less than that given by Smith and Guerin (1980) and is likely a result of using a different number of particle size classes than these authors have used.

The sump used in the Lake Dufault simulations has a constant cross sectional area of 5.94 m^2 and a height of 2.43 metres (Flintoff et al., 1985). The steady state particle frequency size distribution of the pulp in the sump is identical to that given for the hydrocyclone inlet stream earlier (cf. Table 5.4) and the initial steady state height in the sump is 0.96 metres. This value comes from "manually" solving the hydrocyclone and pump models for steady state operation.

Steady state flow rates for the simulation were taken as 65.4 tonnes/hour for the fresh ore feed flow rate; 47.8 tonnes/hour for the sump water feed flow rate; and 11.5 tonnes/hour for the rod mill water feed flow rate.

5.3 Open Loop Response

To utilize a control system with the grinding circuit described in the previous section, it is necessary to have a good understanding of the response of the outputs to changes in the inputs to the process. For this work, three inputs, fresh ore feed rate (FOF), sump water feed rate (SWF) and ore hardness (HRD), and two outputs, hydrocyclone overflow size (COS) and sump pulp level (SPL) will be used in designing the control system. Step changes of a predetermined magnitude are introduced to the input variables one at a time, and the open loop response of both outputs are recorded versus time until new steady states are reached. These records are generally known as process reaction curves (PRC) and can be used in a number of different ways to obtain information on the behavior of the process.

Process reaction curves for the hydrocyclone overflow size and the sump pulp level are presented in Figures 5.3 through 5.8. Figure 5.3 illustrates the response of COS as the mass percent passing 100 Tyler mesh to both positive and negative 10% step changes (6.54 tonnes/hour) from steady state (65.4 tonnes/hour). Positive and negative step changes in the SWF of 10% (4.78 tonnes/hour) from the steady state flow rate of 47.8 tonnes/hour produced the COS responses shown in Figure 5.4. The responses in Figures 5.3 and 5.4 indicate that positive and negative step changes of equal magnitude in either input variable produce almost symmetrical responses in the COS. It should be noted that the magnitudes of the step changes used in generating the responses in Figures 5.3 and 5.4 were chosen so that the the new steady state value of the output variable was obtained in a reasonable length of time (about 200 to 250 minutes of simulated time). As well, larger step changes in either the FOF or the SWF will cause the sump to overflow or run dry under open loop conditions, giving unrepresentative results. The response curves shown in Figure 5.5 present the reaction of the COS to 50% positive and negative step

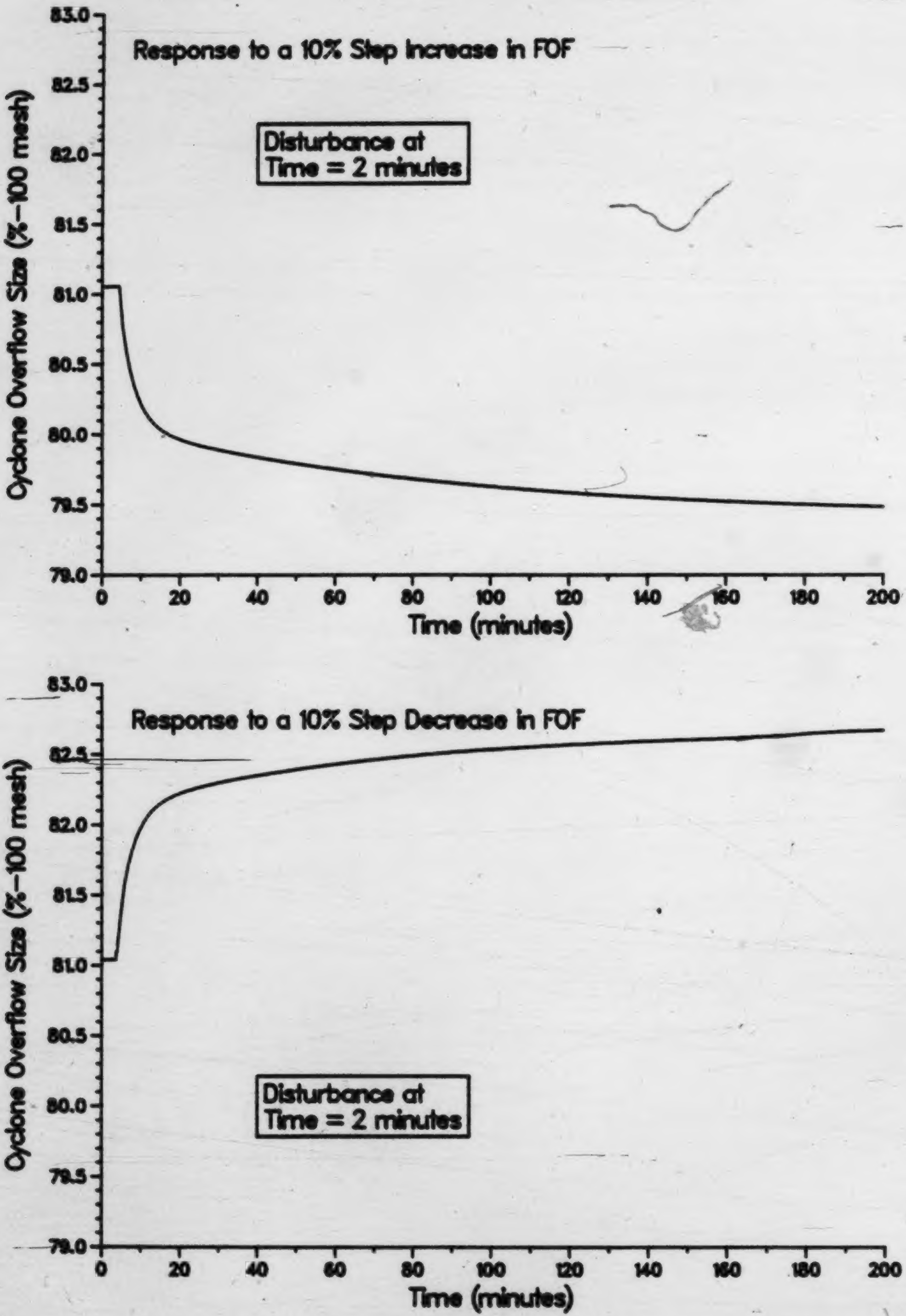


Figure 5.3: Open Loop Response of COS to a ±10% Step in FOF

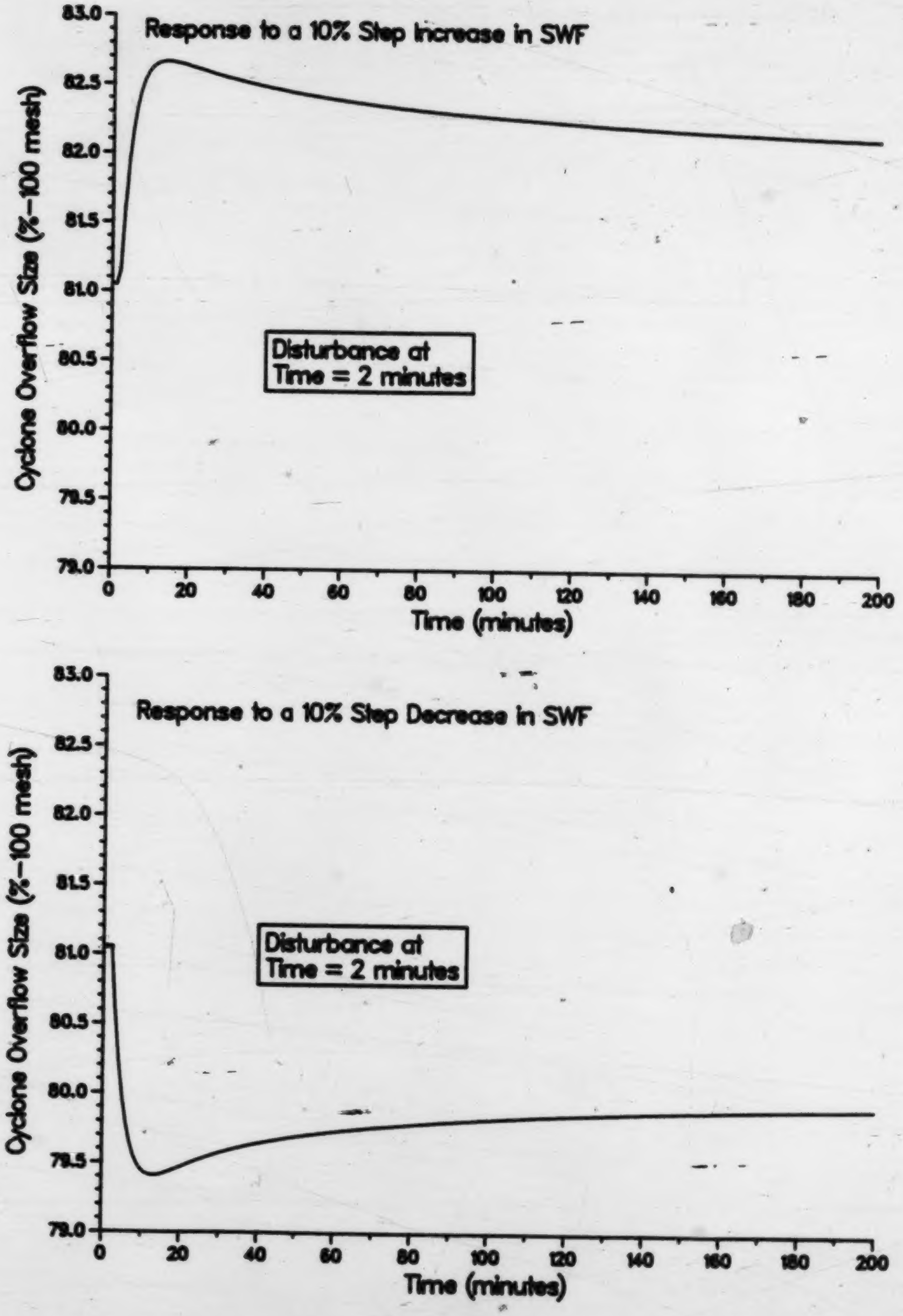


Figure 5.4: Open Loop Response of COS to a ±10% Step in SWF

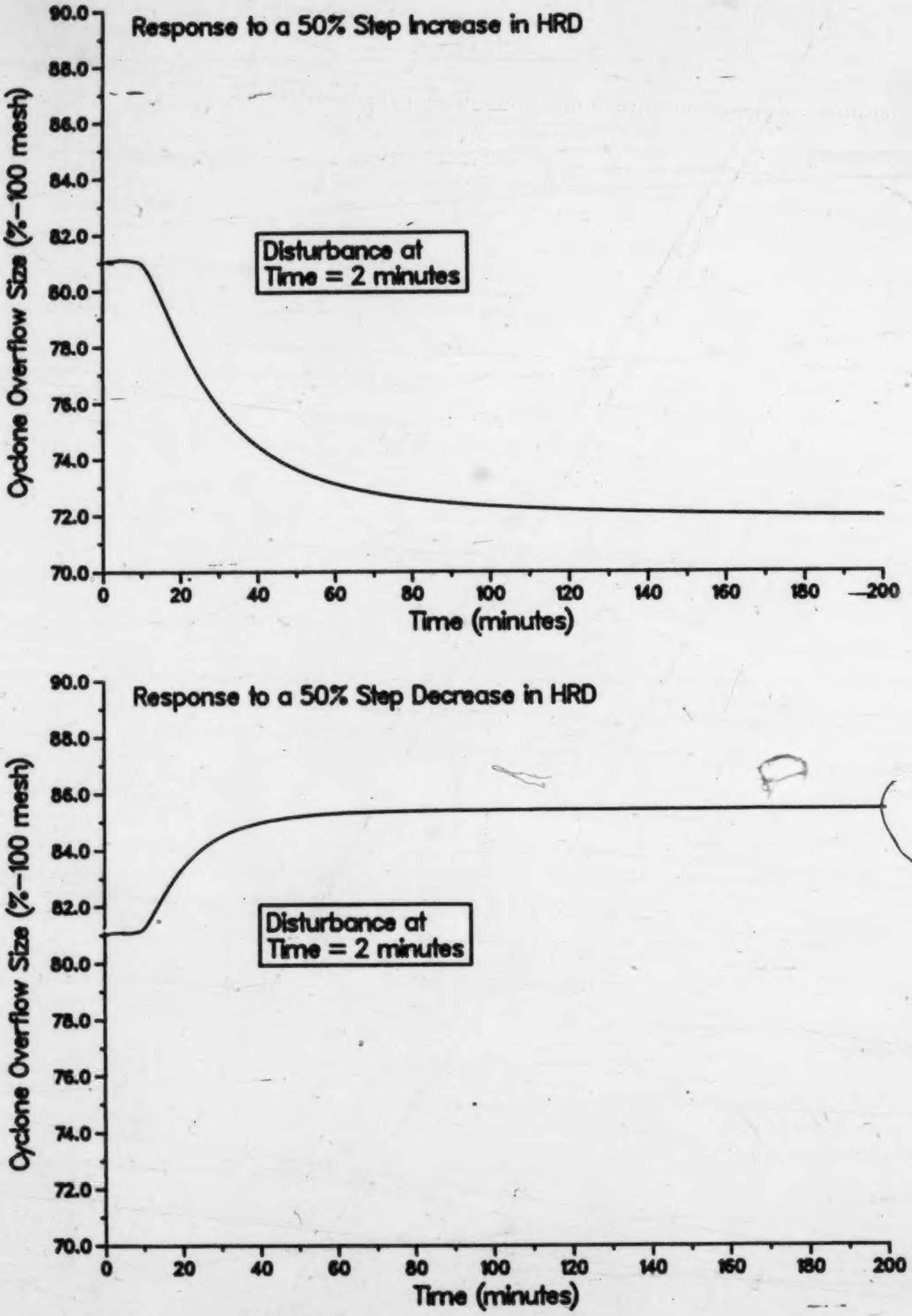


Figure 5.5: Open Loop Response of COS to a ±50% Step in HRD

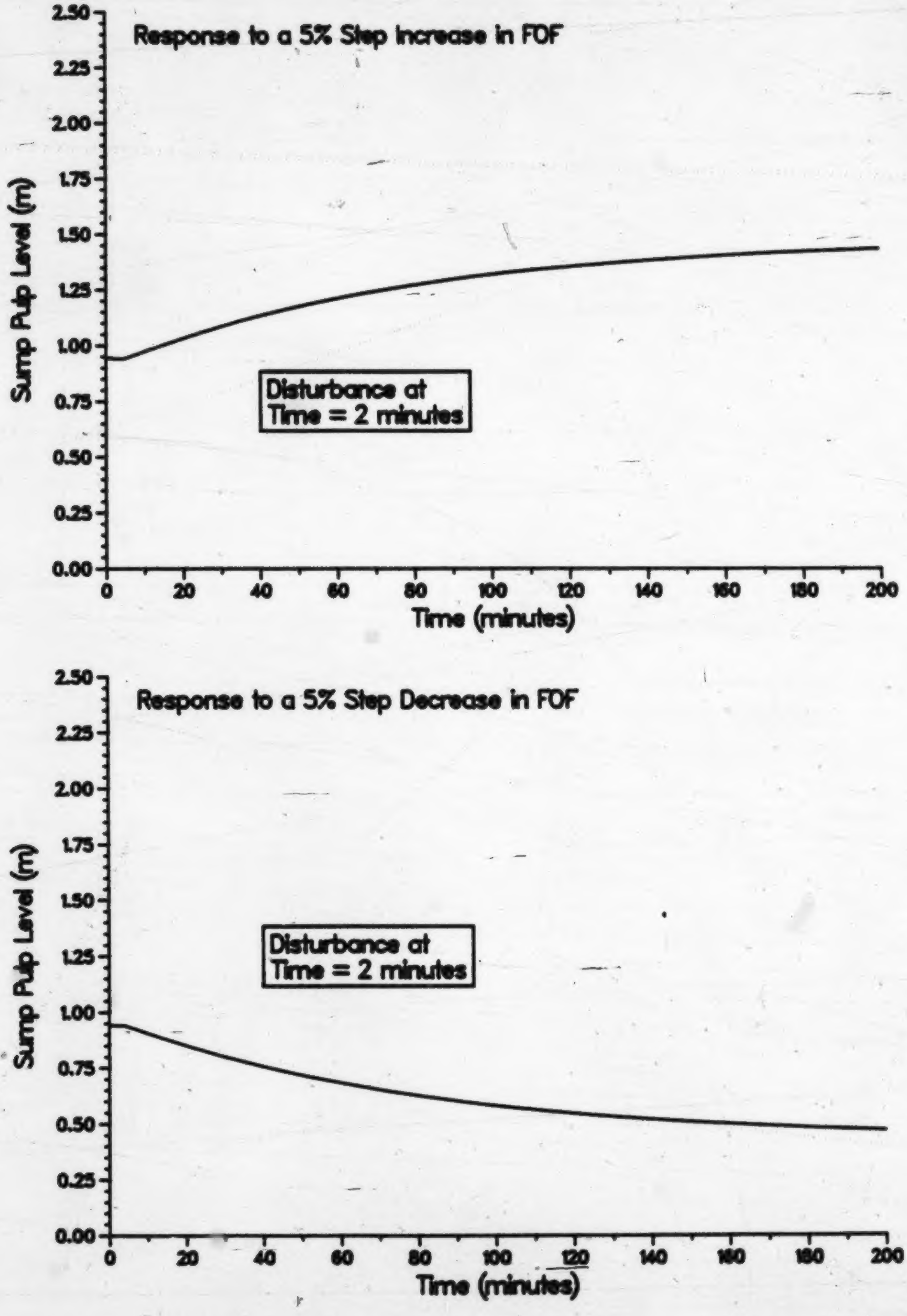


Figure 5.6: Open Loop Response of SPL to a ±5% Step in FOF

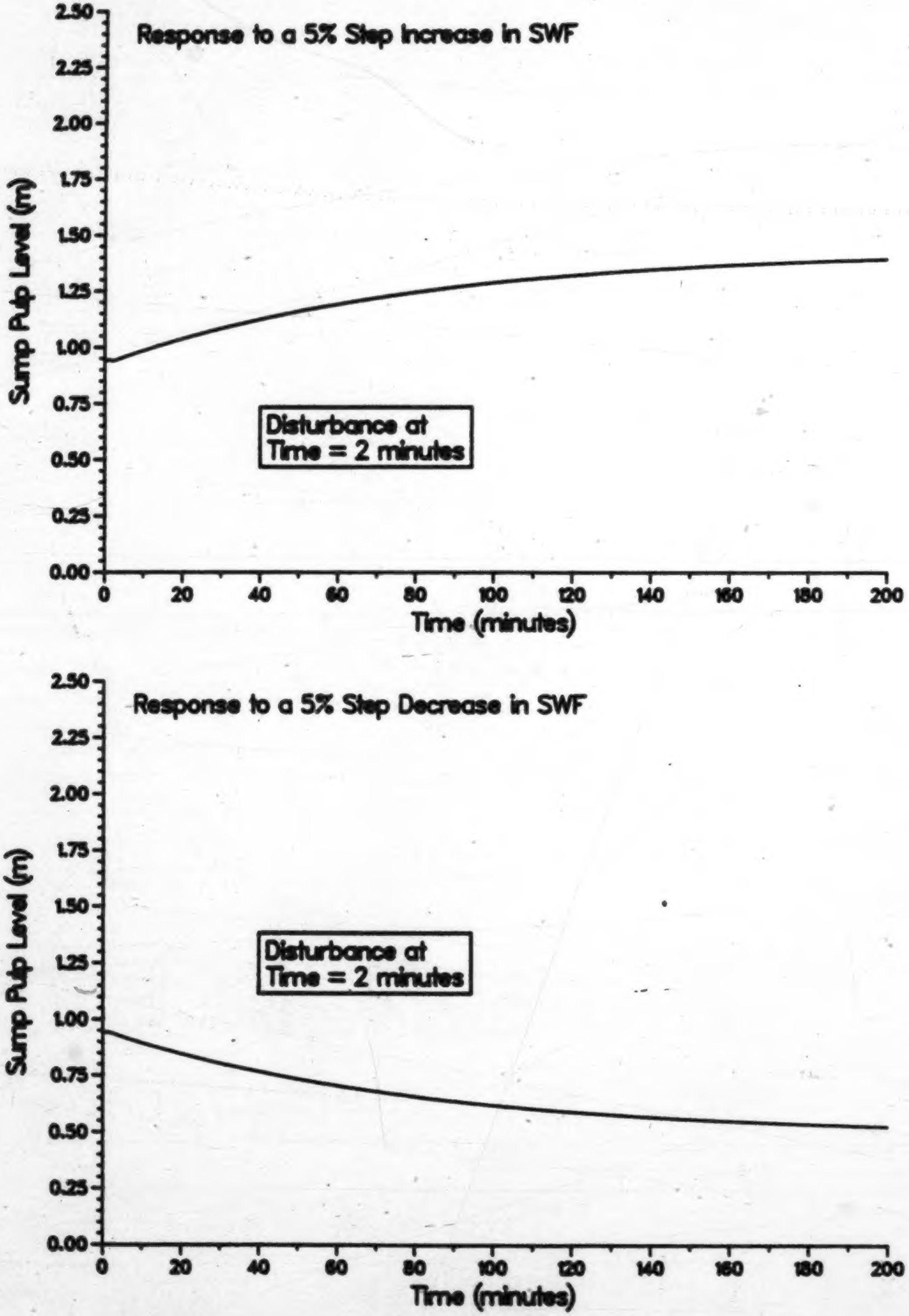


Figure 5.7: Open Loop Response of SPL to a ±5% Step in SWF

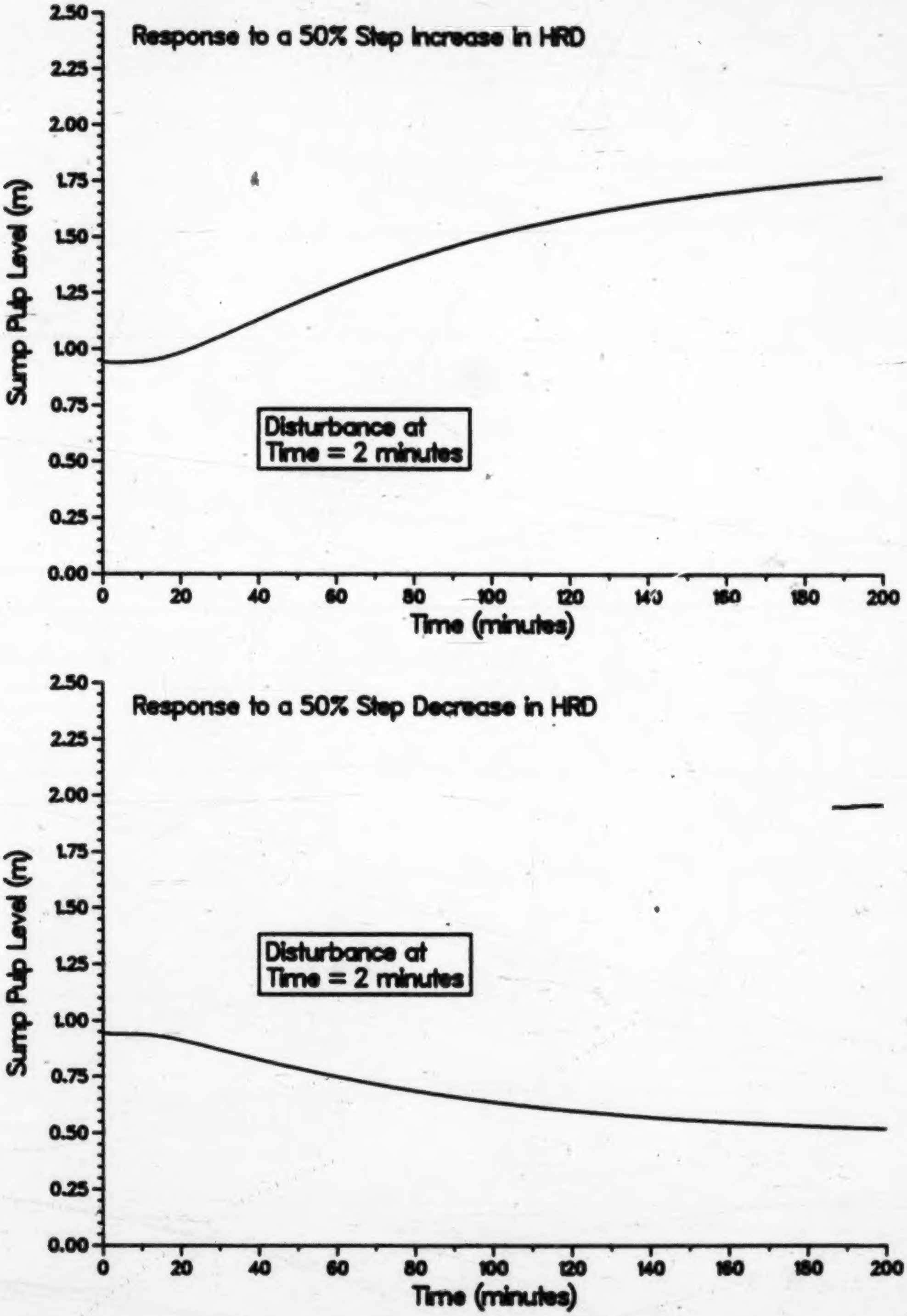


Figure 5.8: Open Loop Response of SPL to a ±50% Step in HRD

changes in ore hardness. This figure clearly shows that the response is dependent on the type of load change which indicates that the process is nonlinear.

Sump pulp level process reaction curves for positive and negative 5% step changes of 3.27 tonnes/hour in the FOF and 2.39 tonnes/hour in the SWF are presented in Figures 5.6 and 5.7. Again the responses appear to be almost symmetrical in both cases. The lower magnitude changes were used because of the slower dynamics involved with the response of the SPL. The new steady states were not reached until a simulated time of about 400 minutes in both cases. The response of the SPL to 50% positive and negative step changes in ore hardness are shown in Figure 5.8. This figure demonstrates that the response of the SPL is nonlinear with respect to ore hardness changes, as was the COS (cf. Figure 5.5). Also, just as for the FOF and COS disturbances, the new steady state value of the sump pulp level was not obtained until about 400 minutes into the simulation.

5.4 Linear Model Representation

The model used to generate the results of the previous section is rather complex. It consists of fifteen ordinary differential equations (ODE's) for each of the two mills coupled through the sump-pump and hydrocyclone models. This results in a forty seventh order system of nonlinear equations that must be solved at each integration step. This is expensive in terms of computer usage and control system design and analysis is aggrivated. A technique known as linearization can be employed which will alleviate this problem by reducing the order of the model. The basis of this technique is to consider the input and output signals of the process, while treating the actual process as a "black box". In this case, the nonlinear model is used to simulate the grinding circuit to produce the required data. Linearization is easily accomplished around a steady state operating point by characterizing the process with transfer functions which relate each input variable to each output variable individually. Since it is common practice to characterize complex dynamic processes with either a first or second order plus time delay transfer function model, and the number of transfer functions required is given by the number of inputs multiplied by the number of outputs, the overall order of the process can be drastically reduced. It should be noted here that the transfer function model that results from this characterization technique will be valid only around the operating point from which it was established. In other words, the further the process operating conditions move from the original steady state operating conditions, the more inaccurate the transfer function model becomes.

In this case, there are three input signals; fresh ore feed rate, sump water feed rate, and ore hardness and two output signals; hydrocyclone overflow size specification and sump pulp level, so six transfer functions are required to model the process. Because ore hardness can not be directly manipulated, it generally acts on the system as a disturbance while the remaining two inputs affect both outputs, naturally leading to a two by two interacting transfer function model of the process. This form of model is shown in Figure 5.9 which is similar to Figure 2.12 except that the appropriate inputs and outputs have been specifically labelled.

For the purposes of this work, a first order plus time delay (FOPTD) transfer function model is considered to be adequate because the initial dynamics are of primary

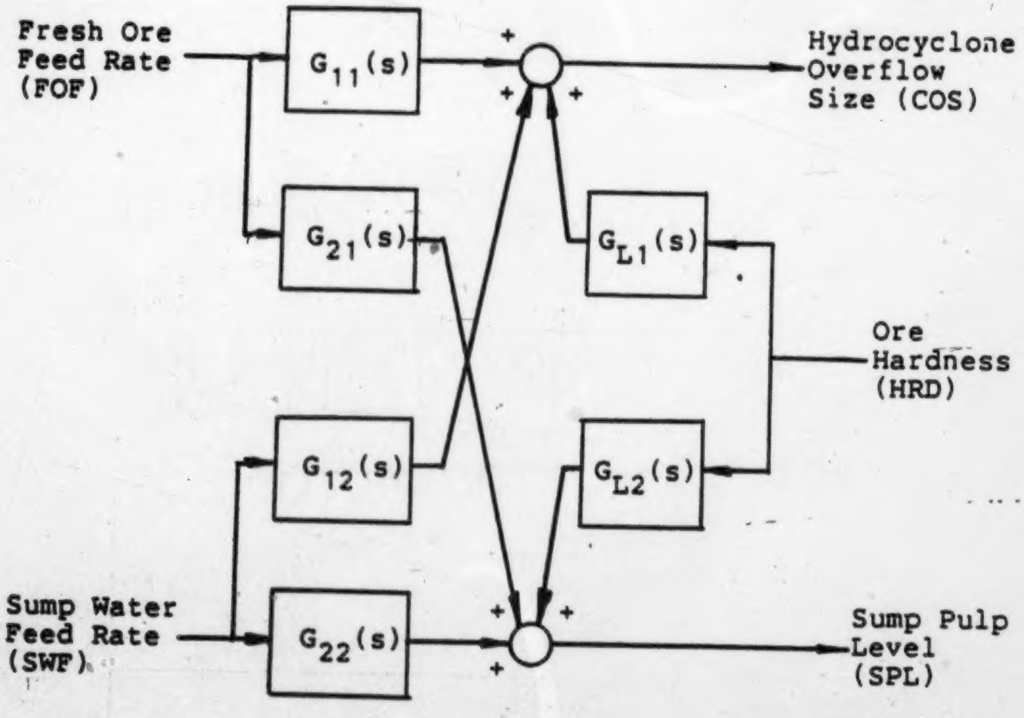

Fresh Ore Feed Rate (FOF)
G11(s)
G21(s)
G12(s)
G22(s)
GL1(s)
GL2(s)
+
+
+
+
+
+
Hydrocyclone Overflow Size (COS)
Ore Hardness (HRD)
Sump Water Feed Rate (SWF)
Sump Pulp Level (SPL)

Figure 5.9: A Block Diagram of a 2x2 Interacting System for the Lake Dufault Grinding Circuit

importance. That is, the process will be modelled using transfer functions of the form:

$$G(s) = \frac{Ke^{-T_d s}}{\tau s+1} \qquad (5.2)$$

There are several well known methods of characterization based on open loop process reaction curves generated by subjecting the process to step input changes. One procedure is to numerically fit a FOPTD transfer function to the curve. This requires a digitized version of the response curve and some sophisticated mathematical analysis which can be very time consuming. More commonly, a graphical analysis technique is used. Smith and Corripio (1985) have investigated several of these procedures and recommend using the technique designated as "FIT 3" because of its reproducibility. In this method, the time delay (T_d) and time constant (τ) in equation (5.2) are chosen so that the model and process responses correspond exactly at times ($t_1=t_o+1/3*\tau$) and ($t_2=t_o+\tau$). These two points typically fall in the zone of highest response change rate and for a true first order system are located on the process reaction curve at 28.3% and 63.2% of the ultimate process response. Figure 5.10 shows the appropriate points graphically. To calculate the time constant and time delay, the following equations are used:

$$\tau = \frac{3}{2}(t_2-t_1) \qquad (5.3)$$

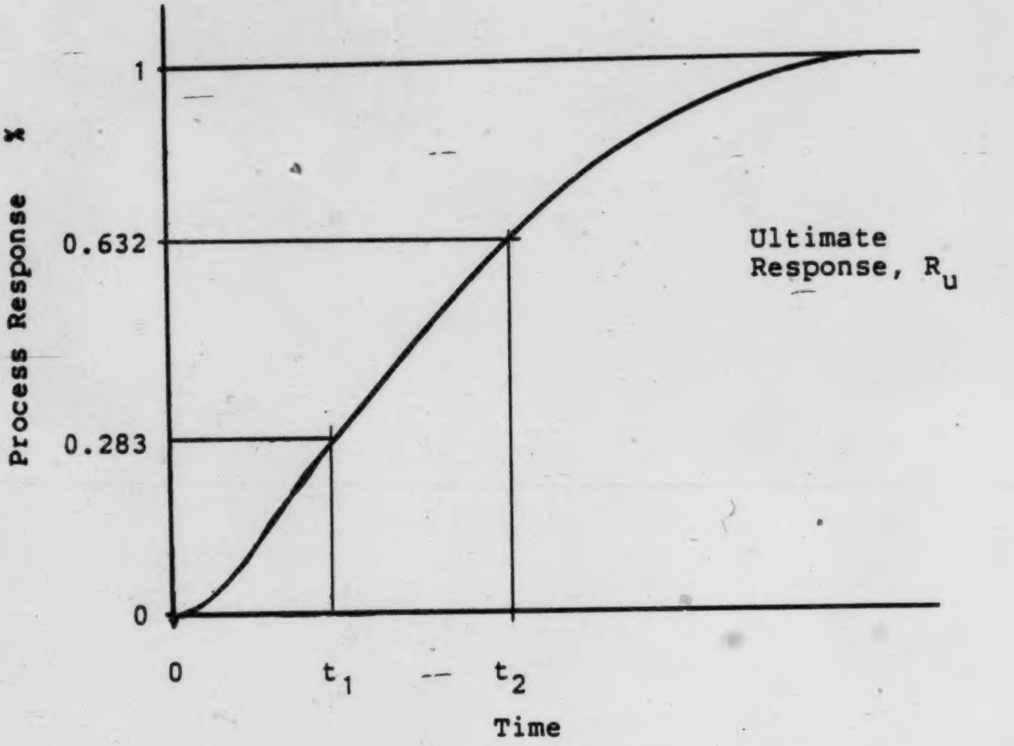


Figure 5.10: A Typical Process Reaction Curve Showing the First Order Plus Time Delay Transfer Function Characterization Points

$$T_d = t_2 - \tau \tag{5.4}$$

The process gain is calculated from the ratio of the ultimate response to the magnitude of the step input:

$$K = B_u / M_s \tag{5.5}$$

5.4.1 The Linear Model

The open loop responses presented in section 5.3 provide all the process reaction curves required to derive the model. Figure 5.11 repeats the negative FOF disturbance response from Figure 5.3 while Figures 5.12 through 5.16 repeat the positive disturbance responses given in Figures 5.4 through 5.8 (Section 5.3) with the appropriate points marked on them. The steady state process information and ultimate responses are also indicated on these figures.

The transfer function model for fresh ore feed to the hydrocyclone size based on the response in Figure 5.11 is:

$$\frac{COS}{FOF} = \frac{-0.249}{9.81s+1} \tag{5.6}$$

and from Figure 5.12 for the effect of sump water feed rate on hydrocyclone overflow size:

$$\frac{COS}{SWF} = \frac{0.219e^{-0.15s}}{1.32s+1} \tag{5.7}$$

It should be noted that the time delay originally associated with equation (5.6) was negative relative to the markings on the figure, indicating a predictive nature. As this can not be so, the delay has been set to zero. The transfer function for the effect of a change in hardness on the hydrocyclone overflow size based on the responses in

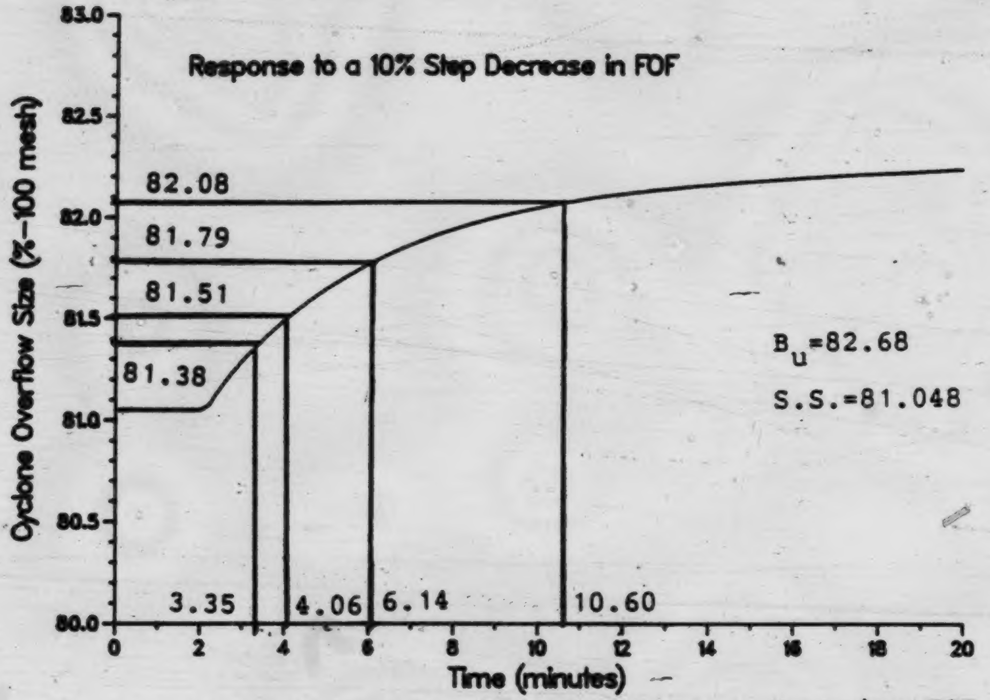


Figure 5.11: Characterization of COS for a Negative FOF Step Disturbance

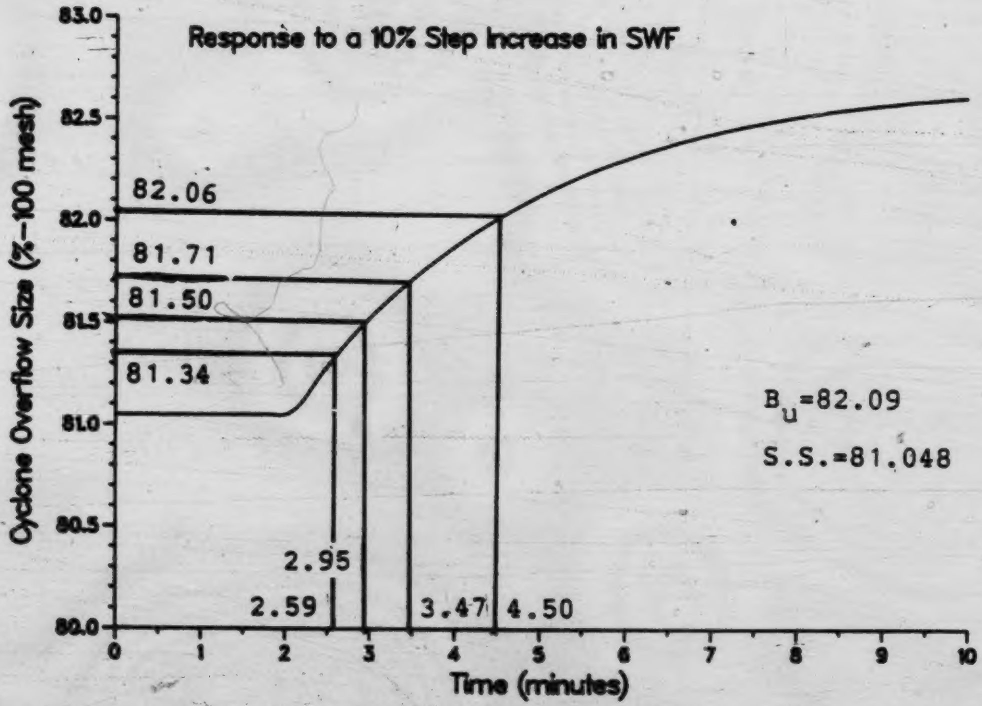


Figure 5.12: Characterization of COS for a Positive SWF Step Disturbance

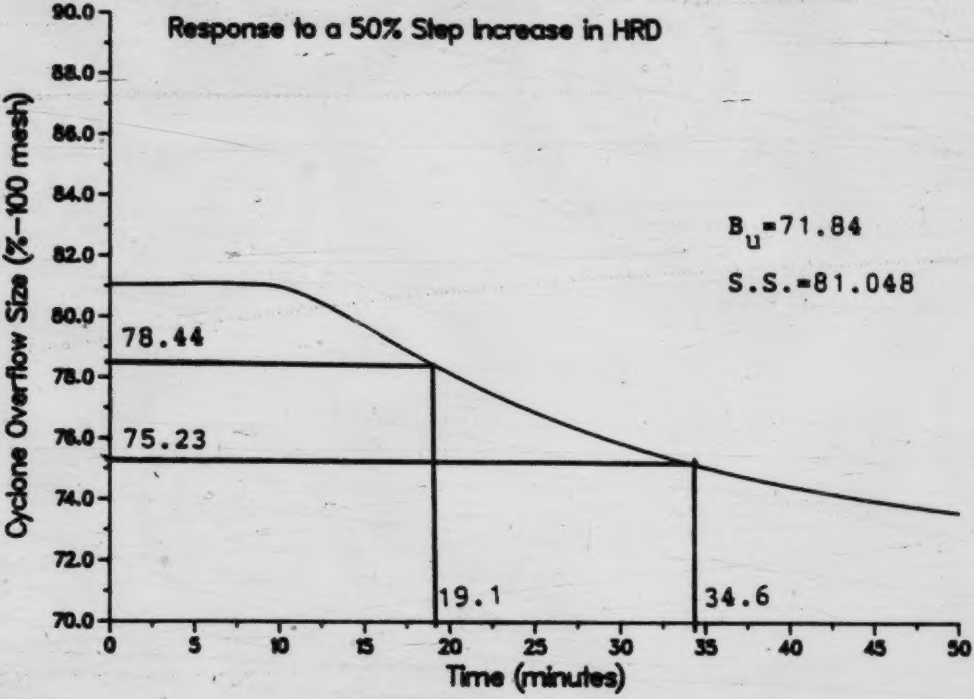


Figure 5.13: Characterization of COS for a Positive HRD Step Disturbance

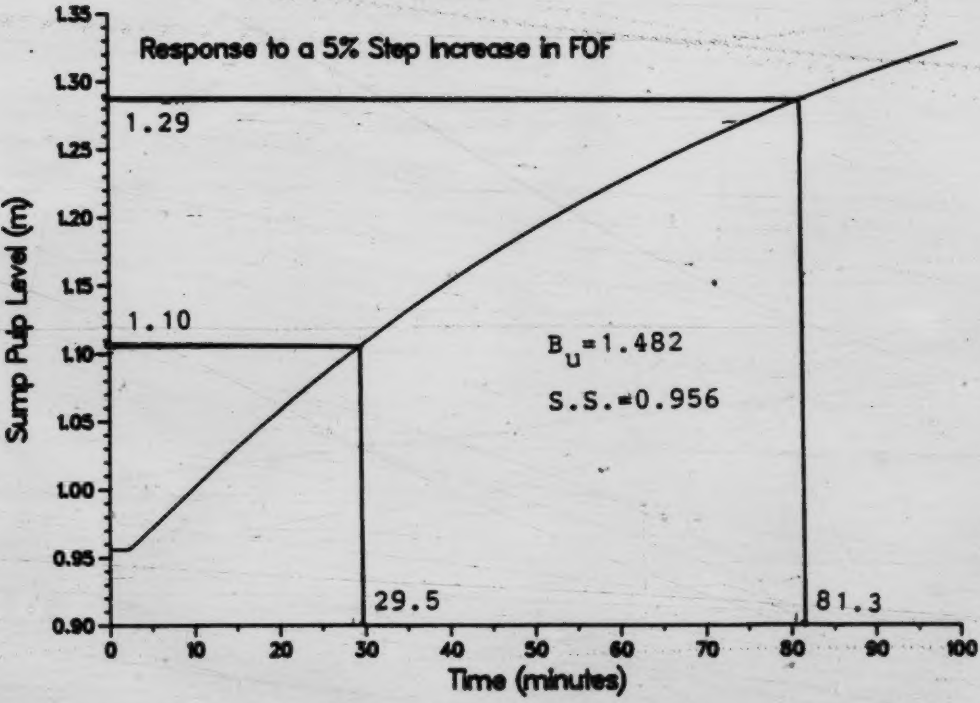


Figure 5.14: Characterization of SPL for a Positive FOF Step Disturbance

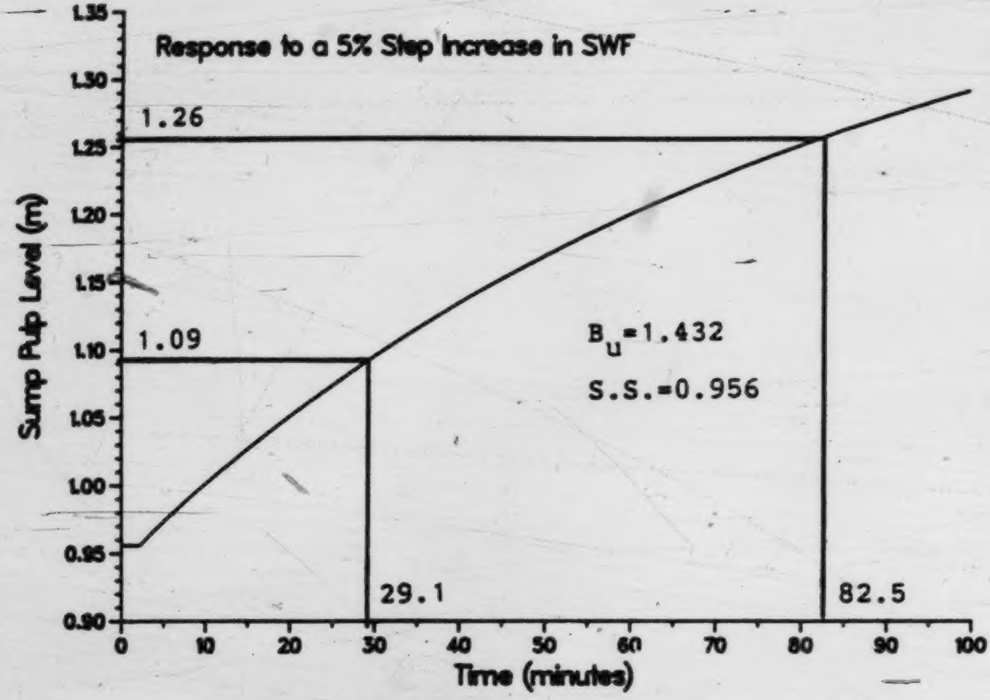


Figure 5.15: Characterization of SPL for a Positive SWF Step Disturbance

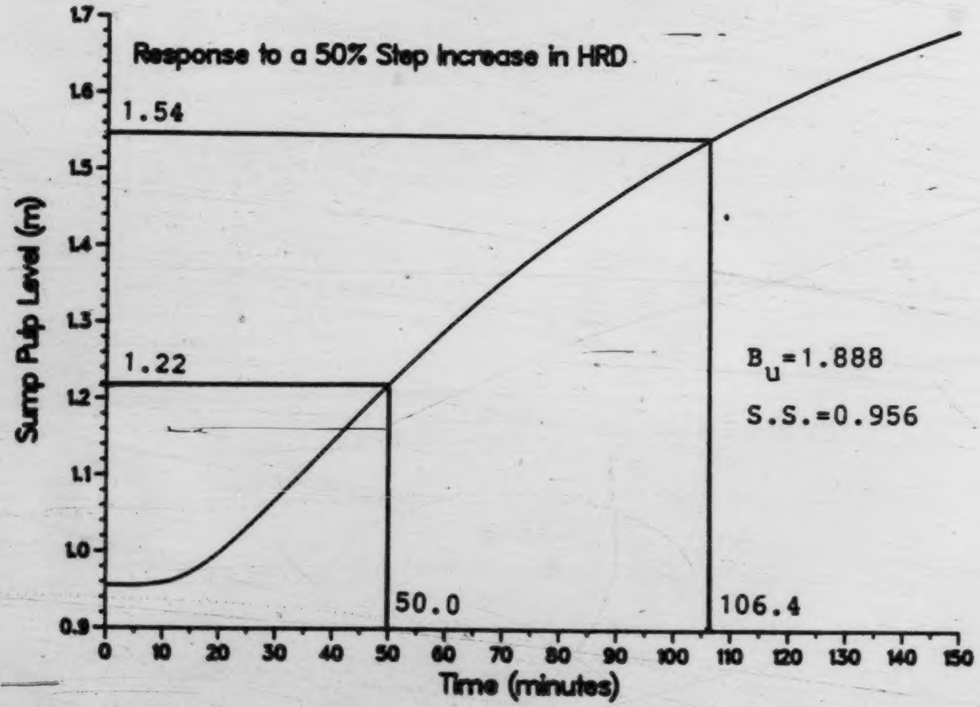


Figure 5.16: Characterization of SPL for a Positive HRD Step Disturbance

Figure 5.13 was found to be:

$$\frac{COS}{HRD} = \frac{-0.184e^{-9.6s}}{22.7s+1} \qquad (5.8)$$

The transfer functions that relate the fresh ore feed rate, the sump water feed rate, and the ore hardness to the sump pulp level, using Figures 5.14, 5.15, and 5.16 respectively, were established as:

$$\frac{SPL}{FOF} = \frac{0.161e^{-1.61s}}{77.6s+1} \qquad (5.9)$$

$$\frac{SPL}{SWF} = \frac{0.199e^{-0.42s}}{80.0s+1} \qquad (5.10)$$

$$\frac{SPL}{HRD} = \frac{0.0186e^{-19.8s}}{84.5s+1} \qquad (5.11)$$

To illustrate that the transfer function model provides a satisfactory representation of the responses calculated from the nonlinear model, the responses for both models subjected to various step inputs are shown in Figures 5.17 through 5.22. It can be seen that the transfer function model agrees reasonably well with the original nonlinear model when subjected to the same step inputs as used in the linear model derivation. However, in some cases, it is

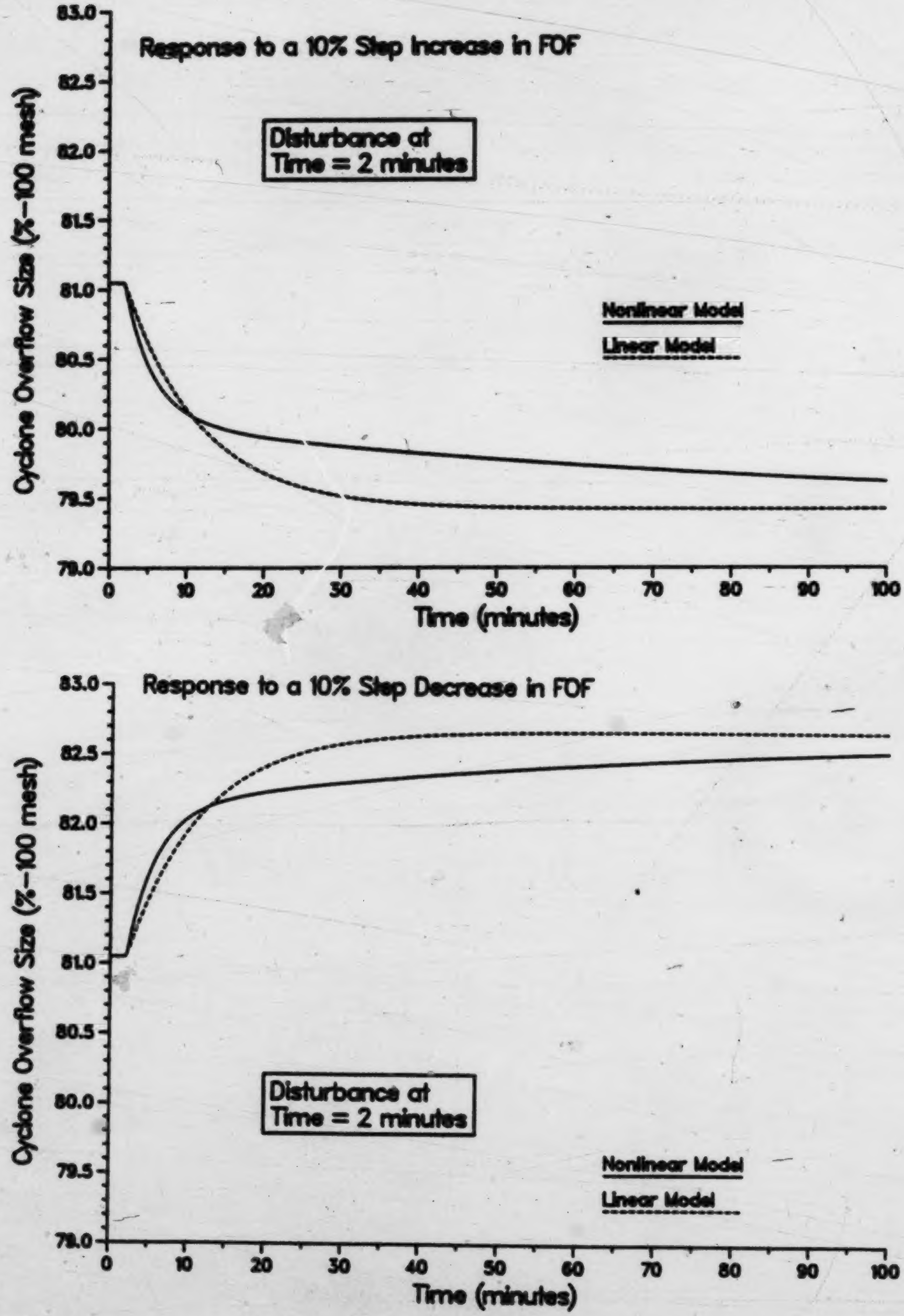
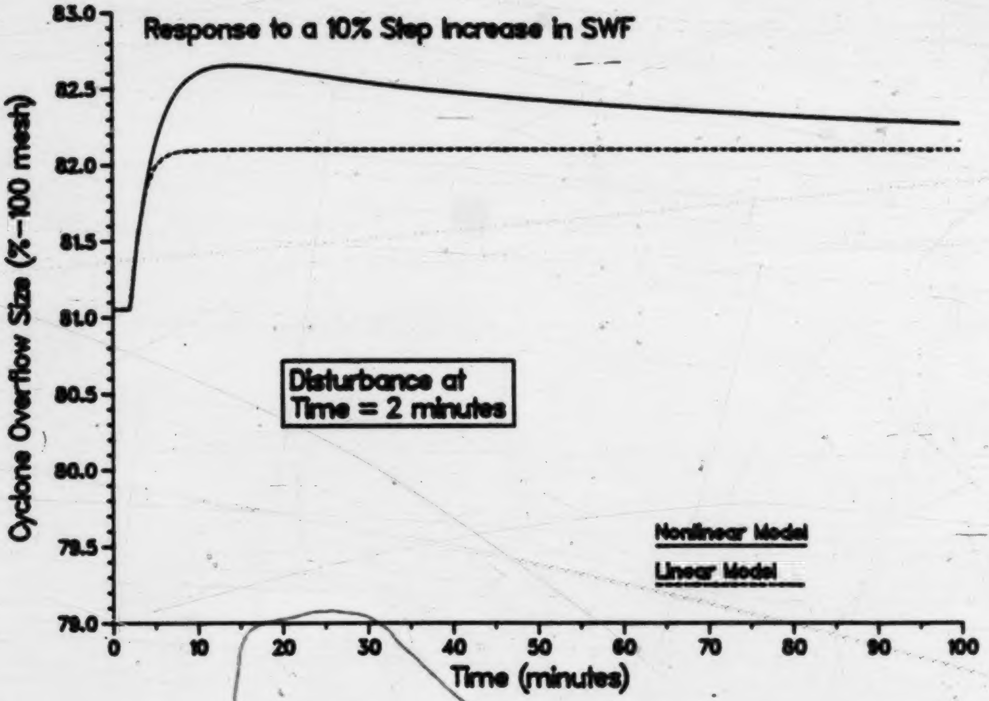


Figure 5.17: Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±10% Step in FOF



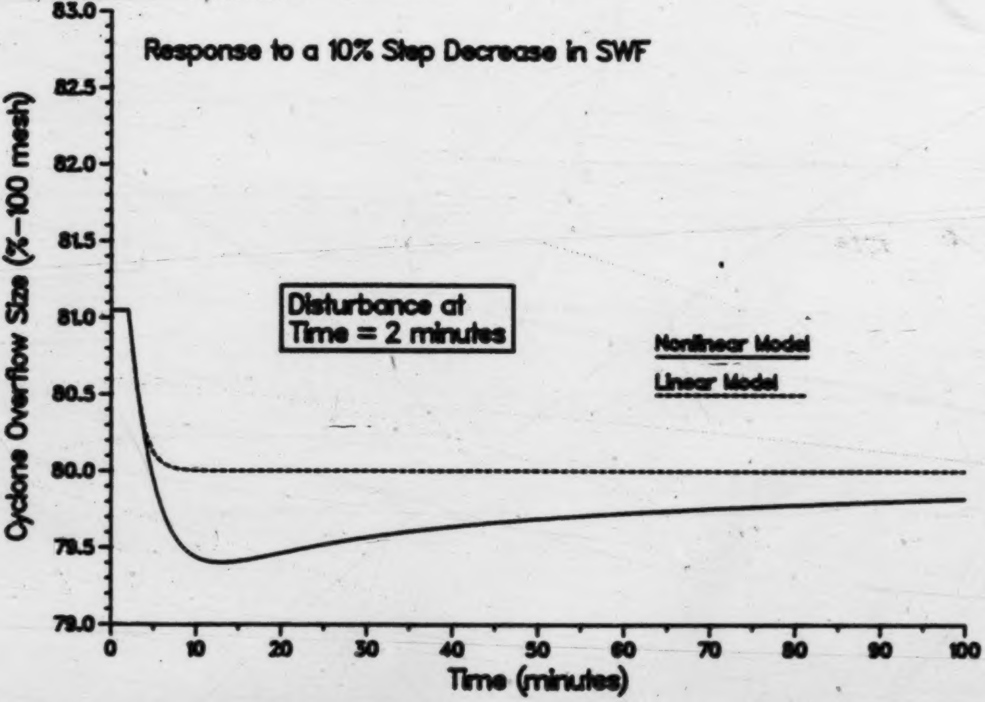


Figure 5.18: Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±10% Step in SWF

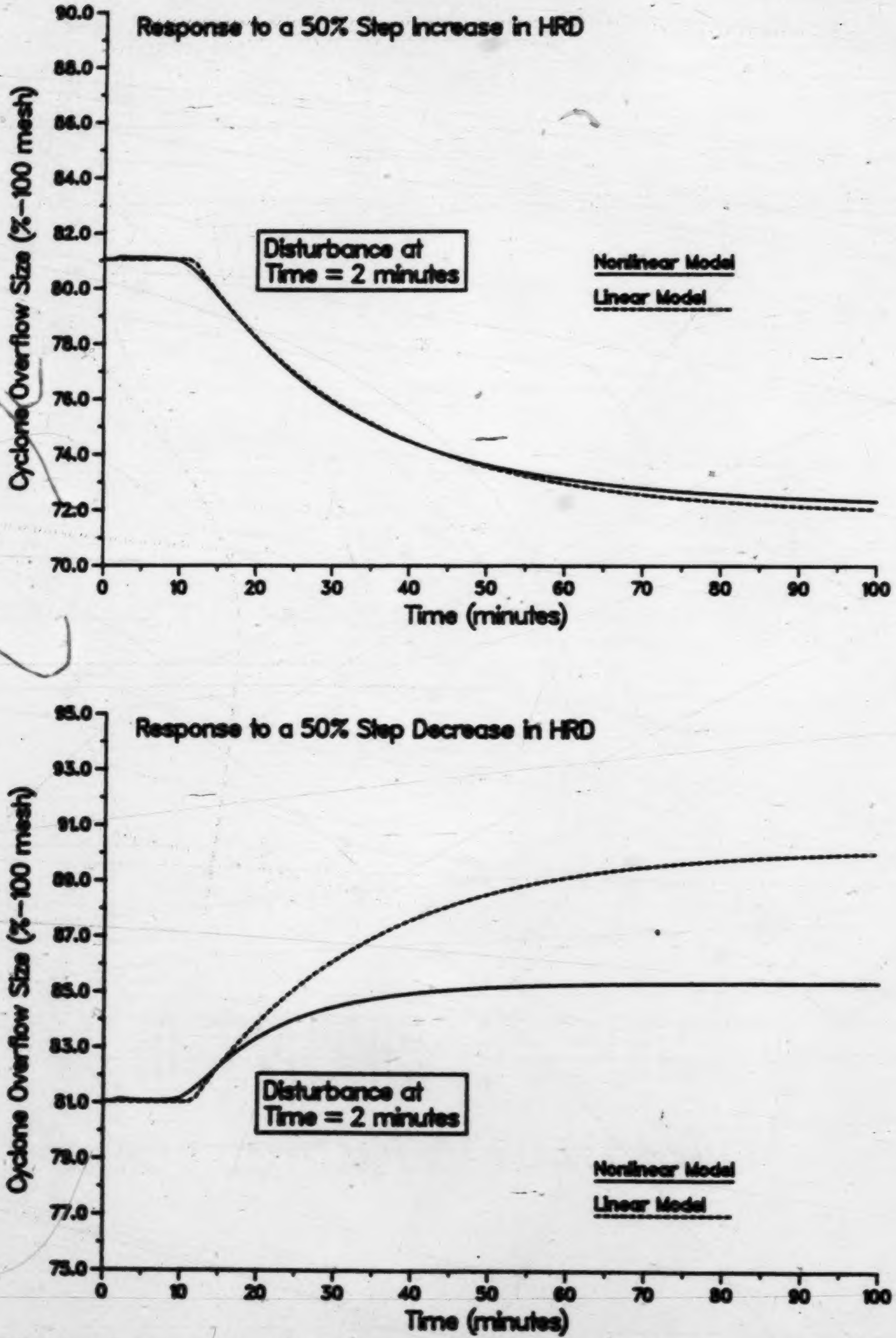


Figure 5.19: Open Loop Response of COS using the Linear and Nonlinear Circuit Models to ±50% Step in HRD

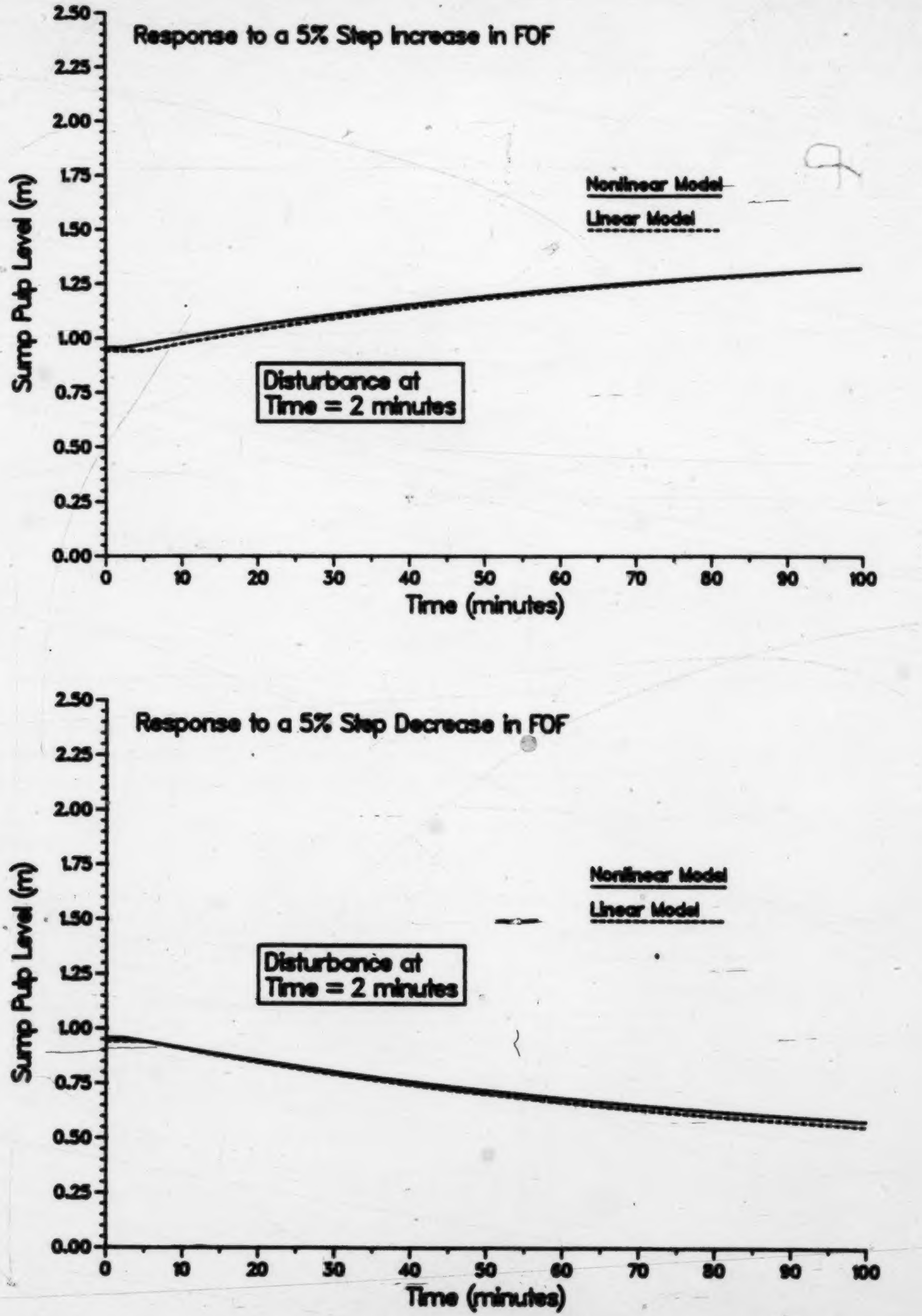
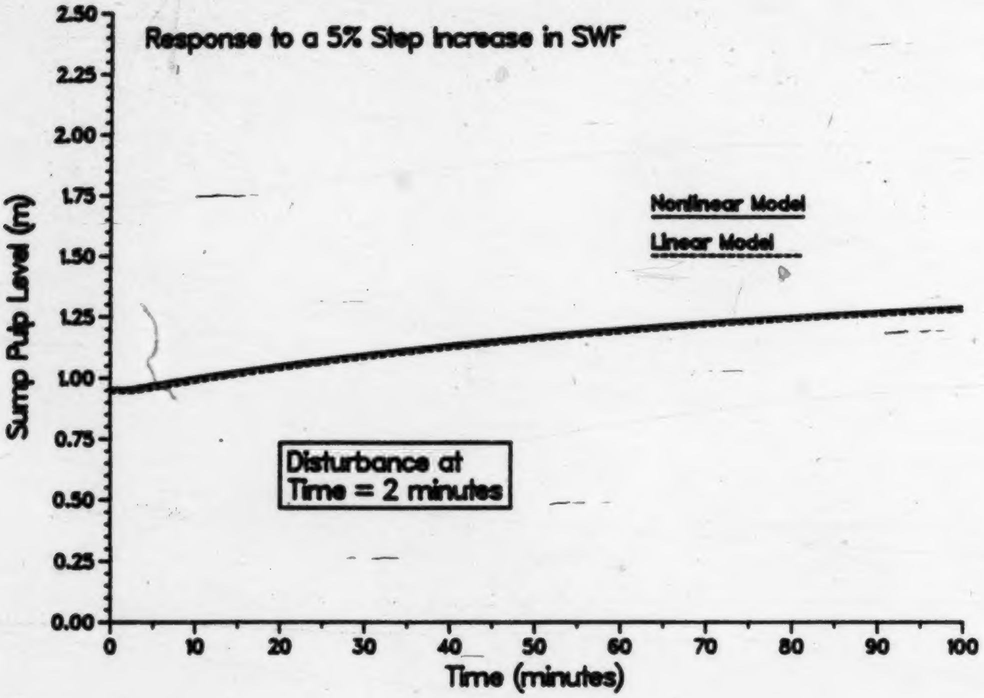


Figure 5.20: Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±5% Step in FOF



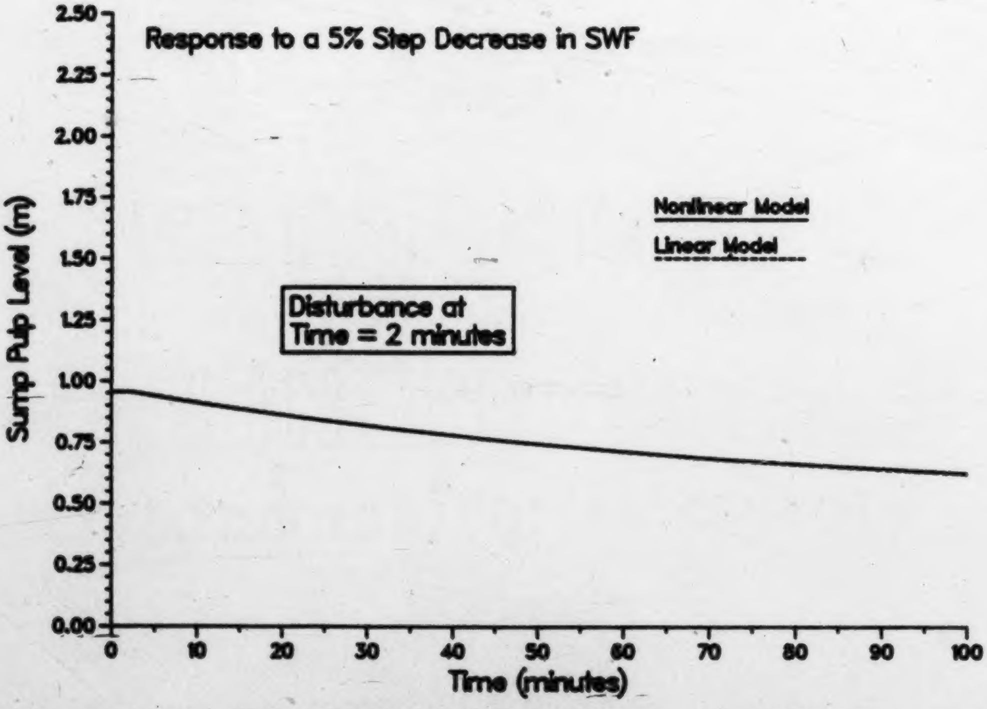


Figure 5.21: Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±5% Step in SWF

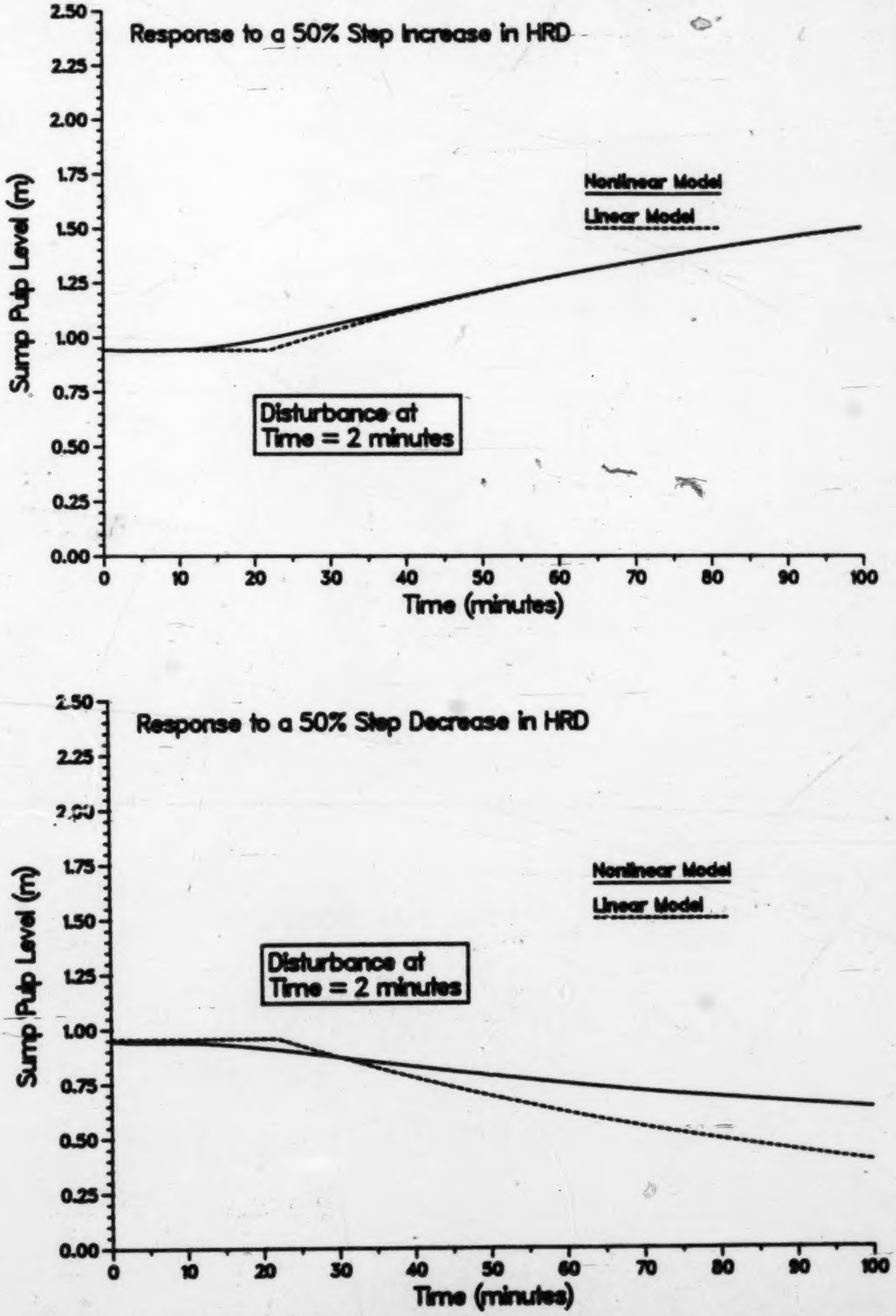


Figure 5.22: Open Loop Response of SPL using the Linear and Nonlinear Circuit Models to ±50% Step in HRD

evident that improvements can be made to more accurately reflect the nonlinear behavior.

5.4.2 Controlled-Manipulated Variable Transfer Function Changes

Although the responses shown in Figures 5.17 and 5.18 do not display much nonlinear behavior, the potential for improvements in the accuracy of the dynamics of the linear model is also evident in the case of the hydrocyclone overflow size response to fresh ore feed rate and sump water feed rate disturbances. In both situations, a higher order model, for example, a second order plus time delay model, would likely produce better results. An alternate approach comes through the realization that the initial dynamics of the aforementioned responses are of primary importance because the input variables are to be manipulated by controllers to resist disturbances caused by ore hardness and to force changes in the initial steady state operation of the output variables. This involves a series of small step changes in the manipulated variables, with the magnitude depending on the deviation of the controlled variables from their desired values. The step changes occur at the beginning of each integration interval and with each step, new trajectories in the controlled variables are started. These trajectories last for the duration of the manipulated variable step, usually one integration time step, or until the controlled variables reach their desired values by the end of a given integration time step. Thus, the accurate representation of the short duration dynamic behavior is of prime concern for either regulatory or servo control simulation.

The transfer functions associated with the controlled variable, hydrocyclone overflow size, can easily be modified to achieve the desired initial response curves. In both cases, the highest rate of change in the output variable (hydrocyclone overflow size) occurs between 0 and 20 minutes. Using the value of the output variable at 20 minutes as the ultimate response for each, the first order transfer functions were found to be:

$$\frac{COS}{FOF} = \frac{-0.179}{4.185s+1} \tag{5.12}$$

$$\frac{COS}{SWF} = \frac{0.337e^{-0.18s}}{2.32s+1} \tag{5.13}$$

The responses predicted by these transfer functions to step changes of ±10% in FOF and SWF are shown in Figures 5.23 and 5.24 as are the responses predicted by the nonlinear model. It can be seen that although the ultimate response of the linear models do not show full agreement, the initial dynamics match well.

Additional tuning of the linear model is also necessary for the transfer functions describing the behavior of the SPL to disturbances in the FOF and the SWF. Considering

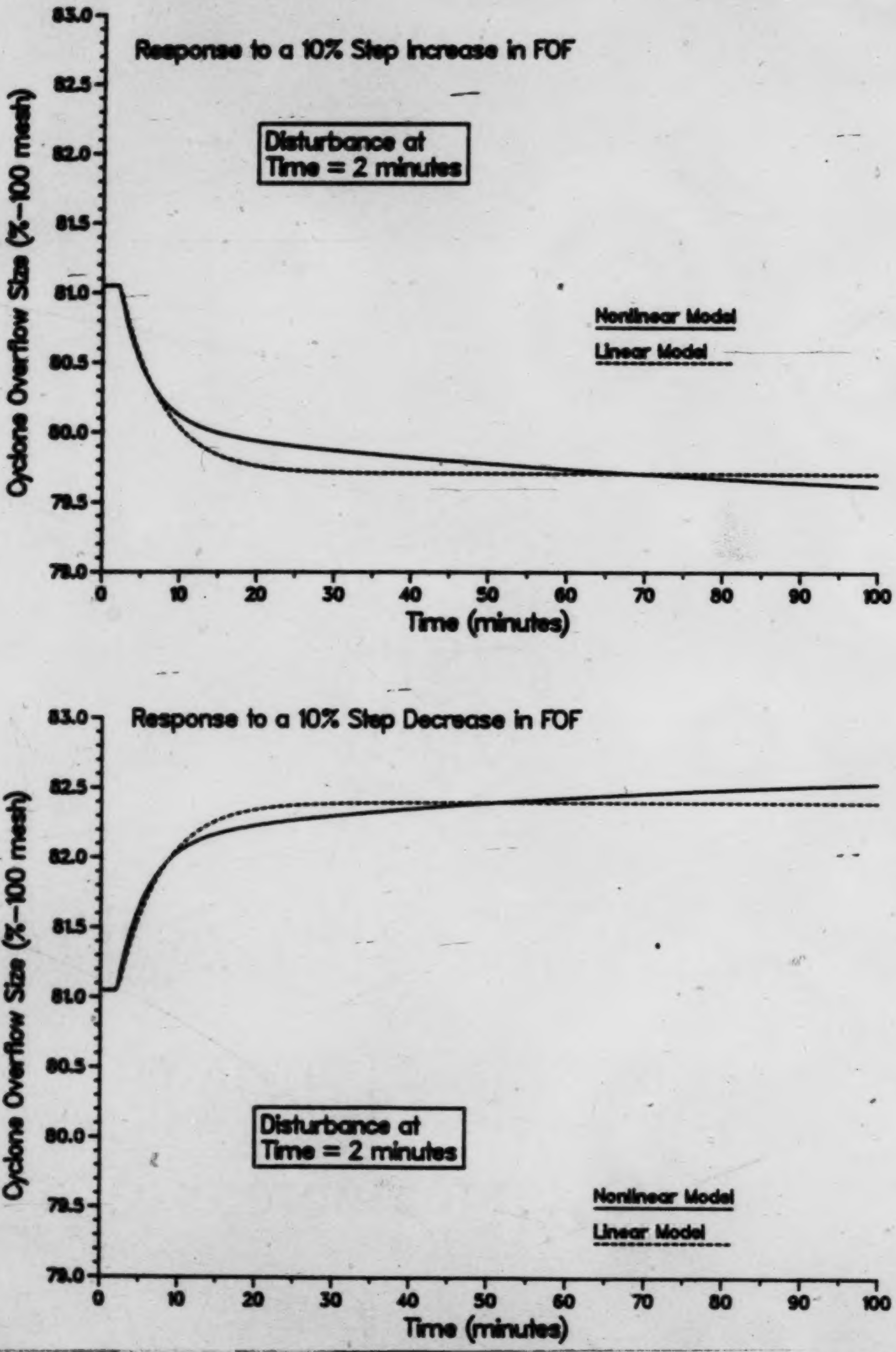
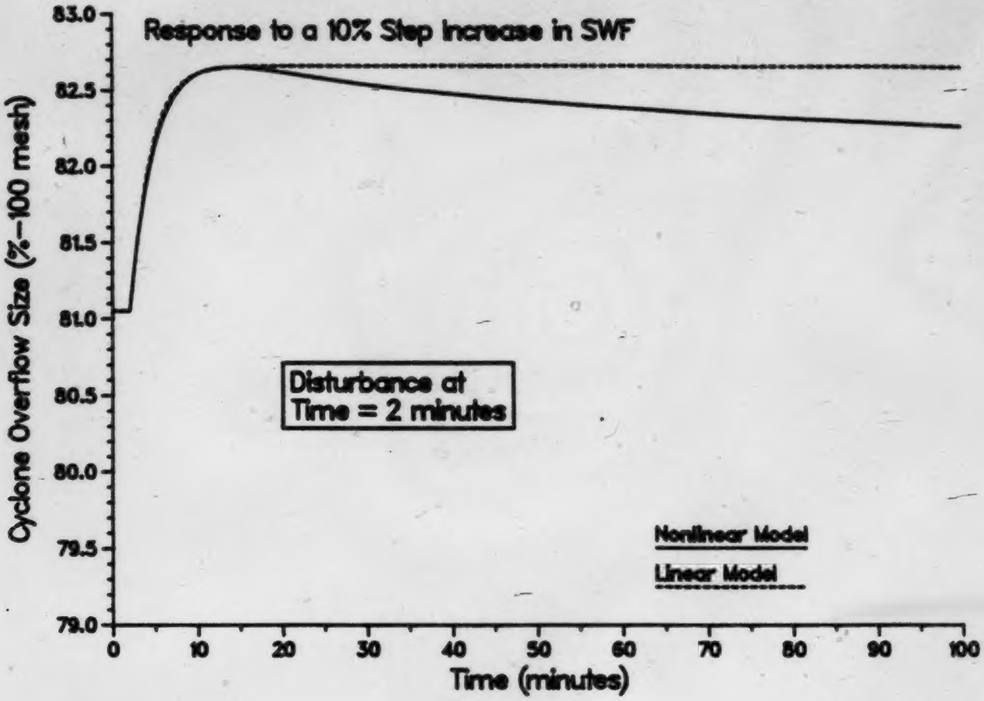


Figure 5.23: Open Loop Response of COS using the Modified Linear Model to ±10% Step Changes in FOF



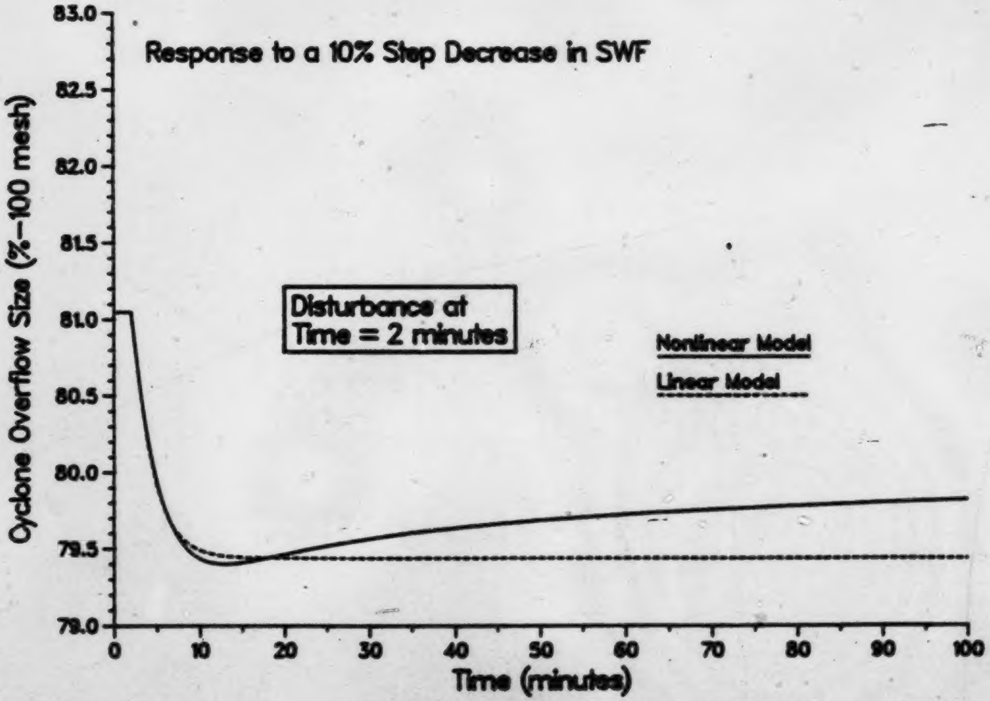


Figure 5.24: Open Loop Response of COS using the Modified Linear Model to ±10% Step Changes in SWF

Figures 5.20 and 5.21, it is obvious that the time delays suggested by equations (5.9) and (5.10) are too long. The time delay for the SPL response to FOF inputs will be reduced by one minute and the delay for SWF inputs will be assumed to be 0.1 minutes.

5.4.3 Ore Hardness Disturbance Transfer Functions Changes

As can be observed from Figures 5.19 and 5.22, the responses of both the COS and the SPL to step changes in ore hardness of the opposite sign are not modelled accurately. This is not a shortcoming of the model, but is a result of compromise in attempting to represent nonlinear behavior with a linear model. Two different approaches can be taken to reduce the effect of this on the overall model. The first involves deriving a seperate transfer function model for use with each type of step input (positive or negative) and a logical switch can be used to decide which particular model should be employed. A second approach is to use an "average" transfer function model based on model constants from the positive and negative distrubances. This latter approach, which is the general practice, is used in this work however it should be noted that this technique introduces modelling error into the responses to both types of inputs.

Figures 5.25 and 5.26 present the responses of the hydrocyclone overflow size and sump pulp level, respectively, when subjected to a 50% decrease in the ore

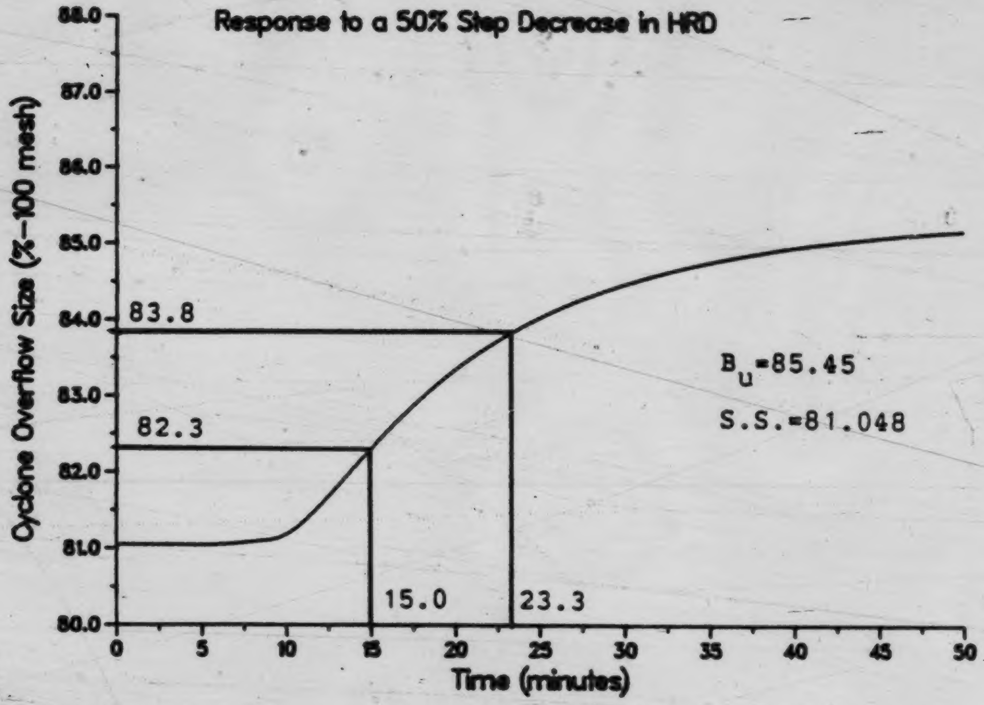


Figure 5.25: Characterization of COS for a Negative HRD Step Disturbance

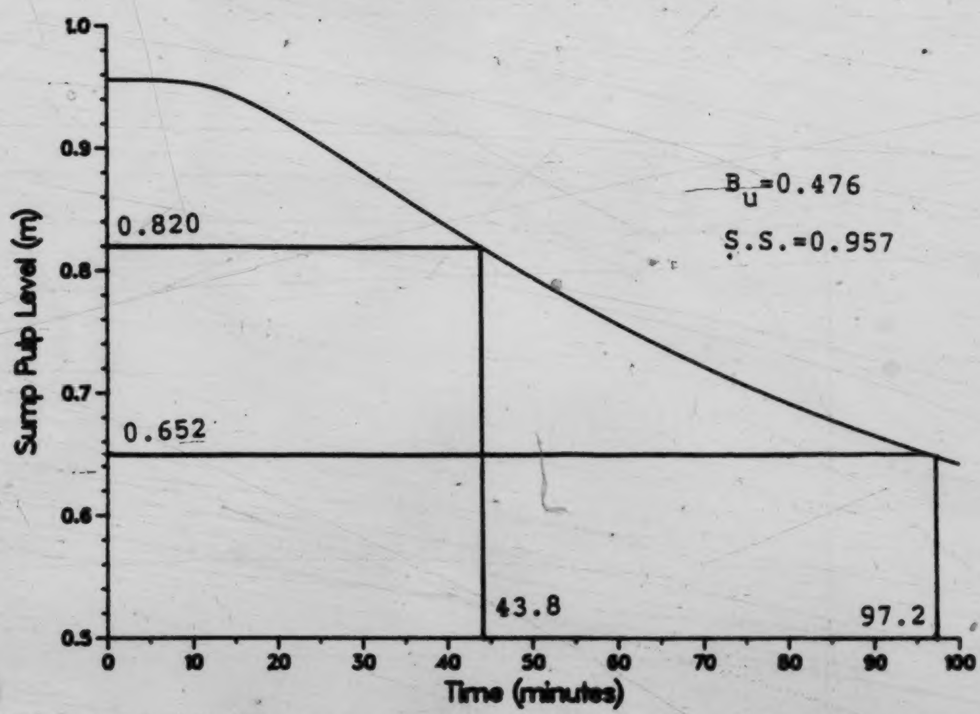


Figure 5.26: Characterization of SPL for a Negative HRD Step Disturbance

hardness, with the points used to derive the negative response transfer functions marked. Using equations (5.3) through (5.5), the first order transfer functions that relate ore hardness to COS and SPL were found to be:

$$\frac{COS}{HRD} = \frac{-0.0881e^{-8.8s}}{12.4s+1} \tag{5.14}$$

$$\frac{SPL}{HRD} = \frac{0.00959e^{-15.1s}}{80.1s+1} \tag{5.15}$$

The average transfer function model is calculated by averaging the constants in equations (5.8) and (5.11) with those in equations (5.14) and (5.15) to yield:

$$\frac{COS}{HRD} = \frac{-0.136e^{-9.2s}}{17.6s+1} \tag{5.16}$$

$$\frac{SPL}{HRD} = \frac{0.0141e^{-17.5s}}{82.3s+1} \tag{5.17}$$

Observing Figures 5.19 and 5.22 once again, it should be clear that further adjustments to the time delays in the disturbance transfer functions should be made in order to more closely match the nonlinear model. Reducing the time delays in equations (5.16) and (5.17) to 6.0 minutes and and 8.0 minutes, respectively, yields:

$$\frac{COS}{HRD} = \frac{-0.136e^{-6.0s}}{17.6s+1} \tag{5.18}$$

$$\frac{SPL}{HRD} = \frac{0.0141e^{-8.0s}}{82.3s+1} \tag{5.19}$$

The responses predicted by these two transfer functions used to model ore hardness disturbances are compared to responses generated by the nonlinear model subjected to both positive and negative step changes in ore hardness in Figures 5.27 and 5.28. It can be seen that the transfer function model responses for either type of step deviate from the nonlinear model response, but the absolute error in both cases is the same.

A summary of the transfer functions used as the model for the subsequent simulations in this chapter is given in Table 5.5 and also shown in the block diagram in Figure 5.29 for convenience and is not meant to imply a particular variable pairing.

5.5 Estimation of PI Controller Constants

Initial controller constants can be estimated by a number of different methods. Most of the more well known procedures are classified as either an on-line procedure such as the Ziegler - Nichols ultimate gain technique or are based on the assumption that the process can be adequately modelled by a first order plus time delay transfer function (the process reaction curve method) and controller constants estimated. Although no one single method can be deemed more satisfactory than another, one method may result in more satisfactory control performance than the other methods for a given process. However, regardless of the method

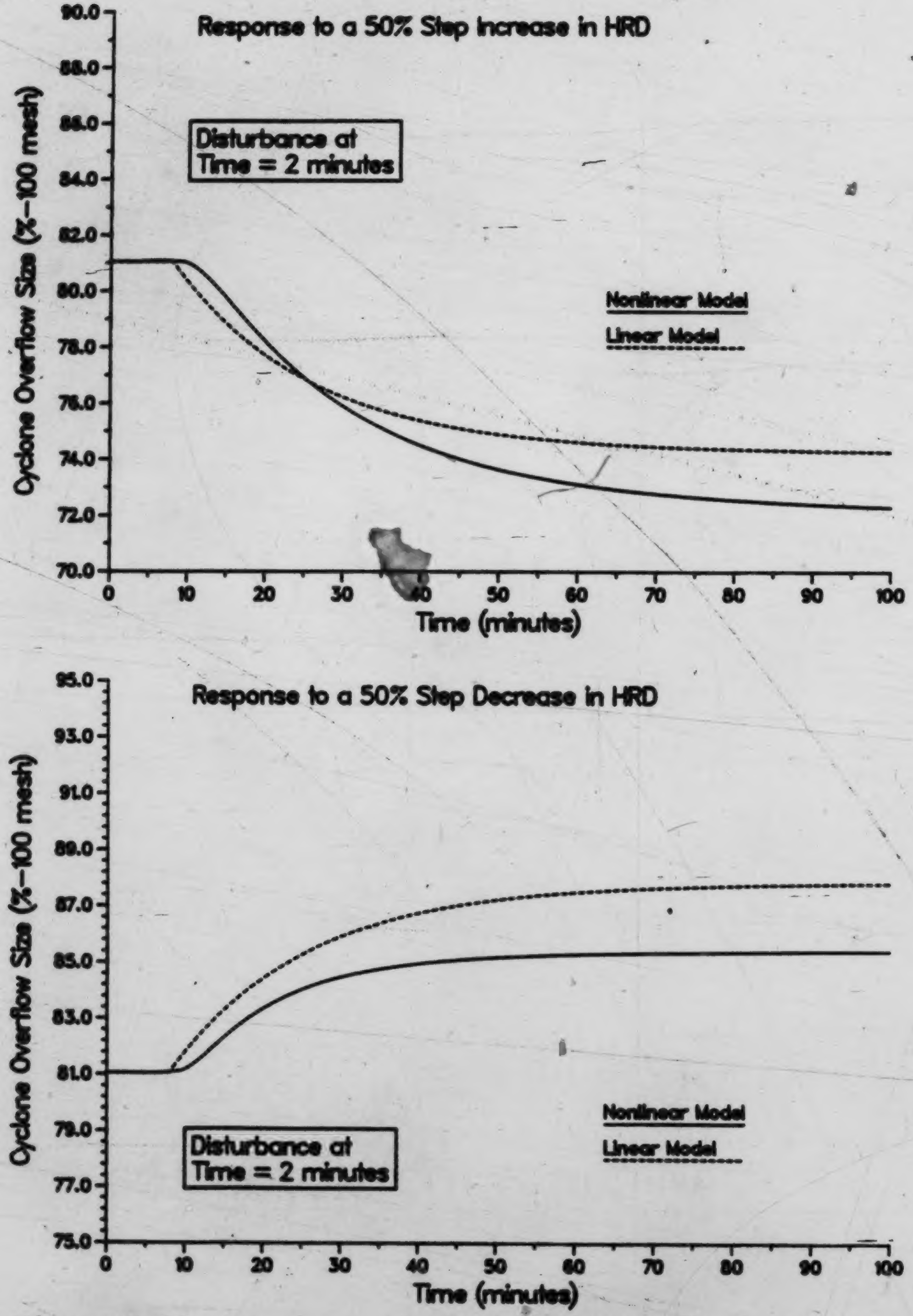


Figure 5.27: Open Loop Response of COS using the Average Linear Circuit Models to ±50% Step in HRD

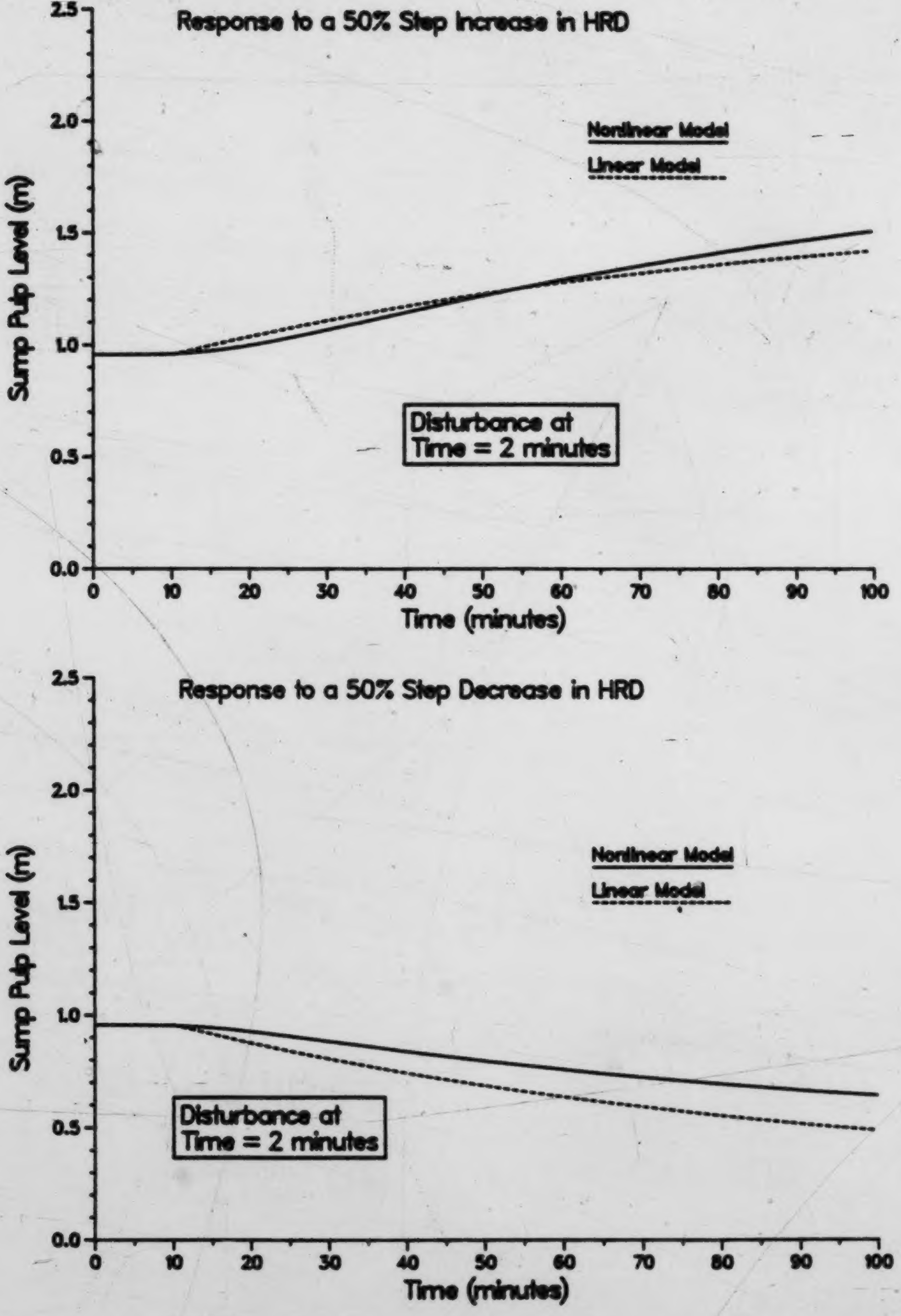


Figure 5.28: Open Loop Response of SPL using the Average Linear Circuit Models to ±50% Step in HRD

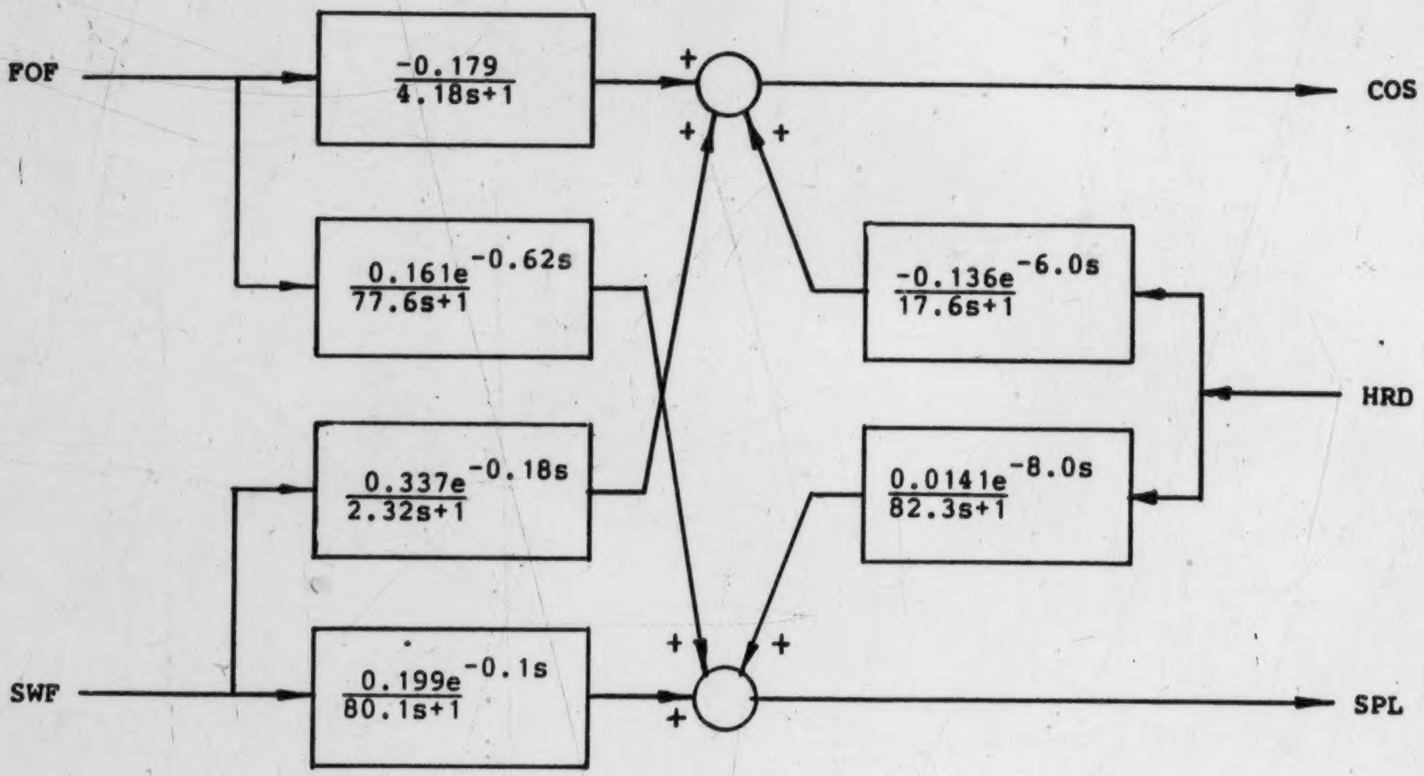

FOF
SWF
HRD
COS
SPL
$\frac{-0.179}{4.18s+1}$
$\frac{0.161e^{-0.62s}}{77.6s+1}$
$\frac{0.337e^{-0.18s}}{2.32s+1}$
$\frac{0.199e^{-0.1s}}{80.1s+1}$
$\frac{-0.136e^{-6.0s}}{17.6s+1}$
$\frac{0.0141e^{-8.0s}}{82.3s+1}$
+
+
+
+
+
+

Figure 5.29: A Block Diagram of the Lake Dufault Grinding Circuit Transfer Function Model

Table 5.5: A Two by Two Transfer Function Model of the Lake Dufault Mineral Grinding Circuit

Manipulated (Input) Variable	Controlled (Output) Variable	
	COS	SPL
FOF	$\frac{-0.179}{4.18s+1}$	$\frac{0.161e^{-0.62s}}{77.6s+1}$
SWF	$\frac{0.337e^{-0.18s}}{2.32s+1}$	$\frac{0.199e^{-0.1s}}{80.1s+1}$
Load HRD	$\frac{-0.136e^{-6.0s}}{17.6s+1}$	$\frac{0.0141e^{-8.0s}}{82.3s+1}$

employed, additional on-line tuning is usually required.

For this work, the process reaction curve method was chosen. It is the simplest and most inexpensive to use from a simulation point of view because only one open loop simulation per control loop to be tuned is needed. The ultimate gain technique would require multiple simulations to obtain the necessary cyclical behavior in the process output.

Controller constants based on single input single output (SISO) system assuming that the open loop system behavior is identical to that produced by a transfer function of the form:

$$G(s) = \frac{Ke^{-T_d s}}{\tau s+1} \qquad (5.20)$$

have been suggested by Cohen and Coon (1953). For proportional integral (PI) controllers, the Cohen and Coon (Coughanowr and Koppel, 1965) recommended values for the controller constants are calculated using the process constants in the following two formulae:

$$K_c = \frac{\tau}{KT_d}\left[\frac{9}{10} + \frac{T_d}{12\tau}\right] \tag{5.21}$$

$$\tau_I = T_d\frac{30 + 3T_d/\tau}{9 + 20T_d/\tau} \tag{5.22}$$

Theoretically, the closed loop response to a setpoint change should have a 1/4 decay ratio with minimum offset, minimum cycling, and minimum area under the response curve when controller constants calculated from these equations are used.

Although equations (5.21) and (5.22) were derived for SISO systems, they can be applied to MIMO systems by assuming no interaction between the individual control loops. In other words, the process is treated as a multiloop noninteracting system and the direct transmission transfer function characteristics are used to estimate the controller constants (i.e. the constants for the $G_{11}(s)$ and $G_{22}(s)$ elements of the process transfer function matrix in the case of a 2x2 system). It is expected that these controller constants will require on-line tuning because process interactions have not been considered.

5.6 Multiloop Control

5.6.1 Type I Control Behavior

It should be clear that with two manipulated variables and two controlled variables, there are two possible variable pairings, generally known as Type I and Type II control system configurations. Figure 5.30 gives a block diagram showing the Type I configuration, including the PI controllers and the model transfer functions. In this case, in loop 1 the fresh ore feed rate (FOF) manipulated variable is paired with the hydrocyclone overflow size (COS) output variable. This particular pairing arises from the fact that

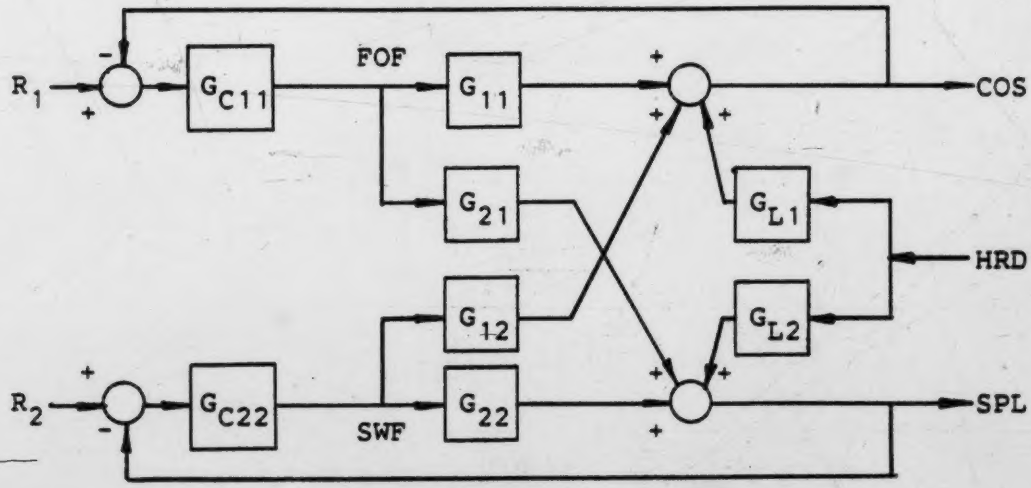


Figure 5.30: Block Diagram Showing the Type I Control System Configuration

the time delays associated with the direct transmission elements of the transfer function model are much shorter in duration (cf. Figure 5.29) than those that result from the Type II pairing. In terms of a process transfer function matrix representation, this control scheme places the dominant time delays in the off diagonal positions which is desirable for achieving satisfactory control performance.

The initial controller tuning constants can be found using equations (5.21) and (5.22) with the gain, time constants, and time delays found for the direct transmission transfer functions of the model. For this control system configuration, these transfer functions are shown as the blocks labelled $G_{11}(s)$ and $G_{22}(s)$ in Figure 5.30 and are given by equations (5.18) and (5.10) in Section 5.4. It immediately becomes apparent that because there is no time delay in loop 1 (cf. $G_{11}(s)$, equation (5.18)), the recommended loop 1 controller gain calculated from equation (5.21) would be infinite and the integral time constant would be zero based on equation (5.22). So to obtain some numerical values, it was assumed that the transfer function had a time delay of 0.01 minutes to give estimated controller constants for loop 1 as:

$K_{C1} = -2100.$

$\tau_{I1} = 0.033$ minutes

while for loop 2, the estimated controller constants are:

$K_{C2} = 3600.$

$\tau_{I2} = 0.33$ minutes

It should be noted that the controller gain required in loop 2 is positive and so a reverse acting controller is required.

Because the normal mode of operation of a grinding circuit is expected to be regulatory, the control system performance will first be investigated for ore hardness disturbances. Using the initial controller estimates given above, the response curves for COS and SPL as calculated by the simulator for a 50% step increase in hardness are shown in Figure 5.31. The controller action is not shown here

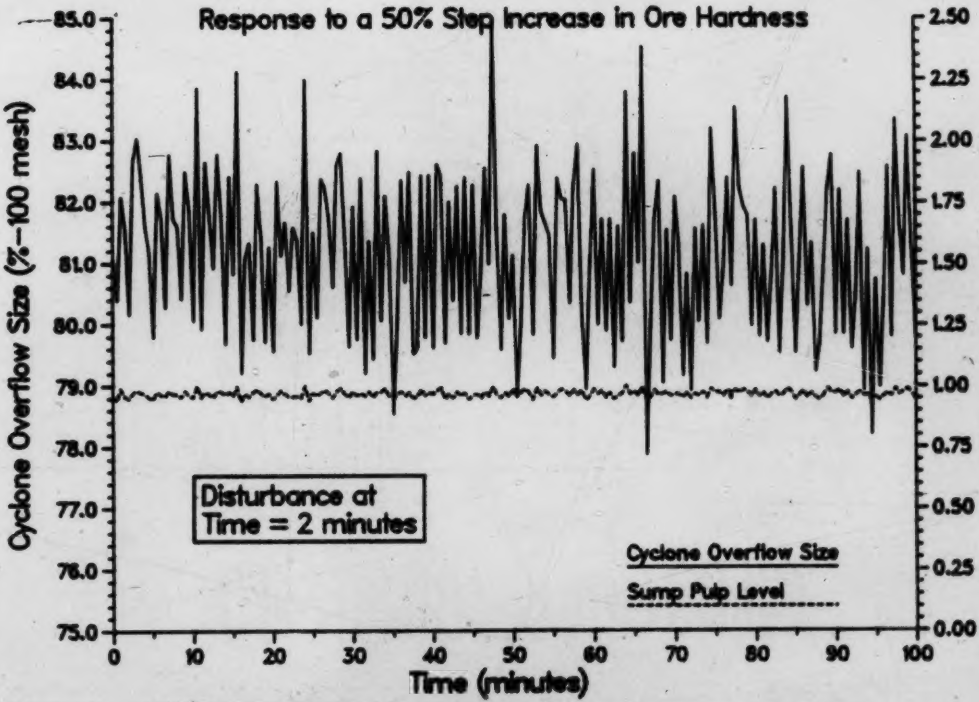


Figure 5.31: Type I Closed Loop Response of the Lake Dufault-Nonlinear Circuit Model for a +50% Step in HRD using Cohen and Coon PI Controller Settings

because it is on-off in nature, and a plot would be meaningless. Clearly, this response behavior is unacceptable, and the required on-off manipulated variable action could not be obtained in practice. The high frequency oscillatory behavior is likely due to process interactions and "tuning" the controllers will produce better response behavior. Figure 5.32 shows the response curves for controllers using tuned controller gains of:

$K_{C1} = -40.0$

$K_{C2} = 82.3$

and the integral time constants of:

$\tau_{I1} = 0.5$ minutes

$\tau_{I2} = 1.38$ minutes

It is evident from the response of the controlled variables and the manipulated variables in Figure 5.32 that the overall system performance has been greatly improved over that shown in Figure 5.31. However, it should be noted that the controller gains had to be reduced from the original estimates. Subjecting this system to a negative 50% step change in ore hardness produces the response shown in Figure 5.33. It is interesting to note in both Figures 5.32 and 5.33, that the control action is slow, reflecting the long process time constants associated with changes in ore hardness because of the closed circuit configuration of the hydrocyclone and ball mill in the Lake Dufault operation. Furthermore, comparison of the responses in Figures 5.32 and 5.33 with the open loop responses in

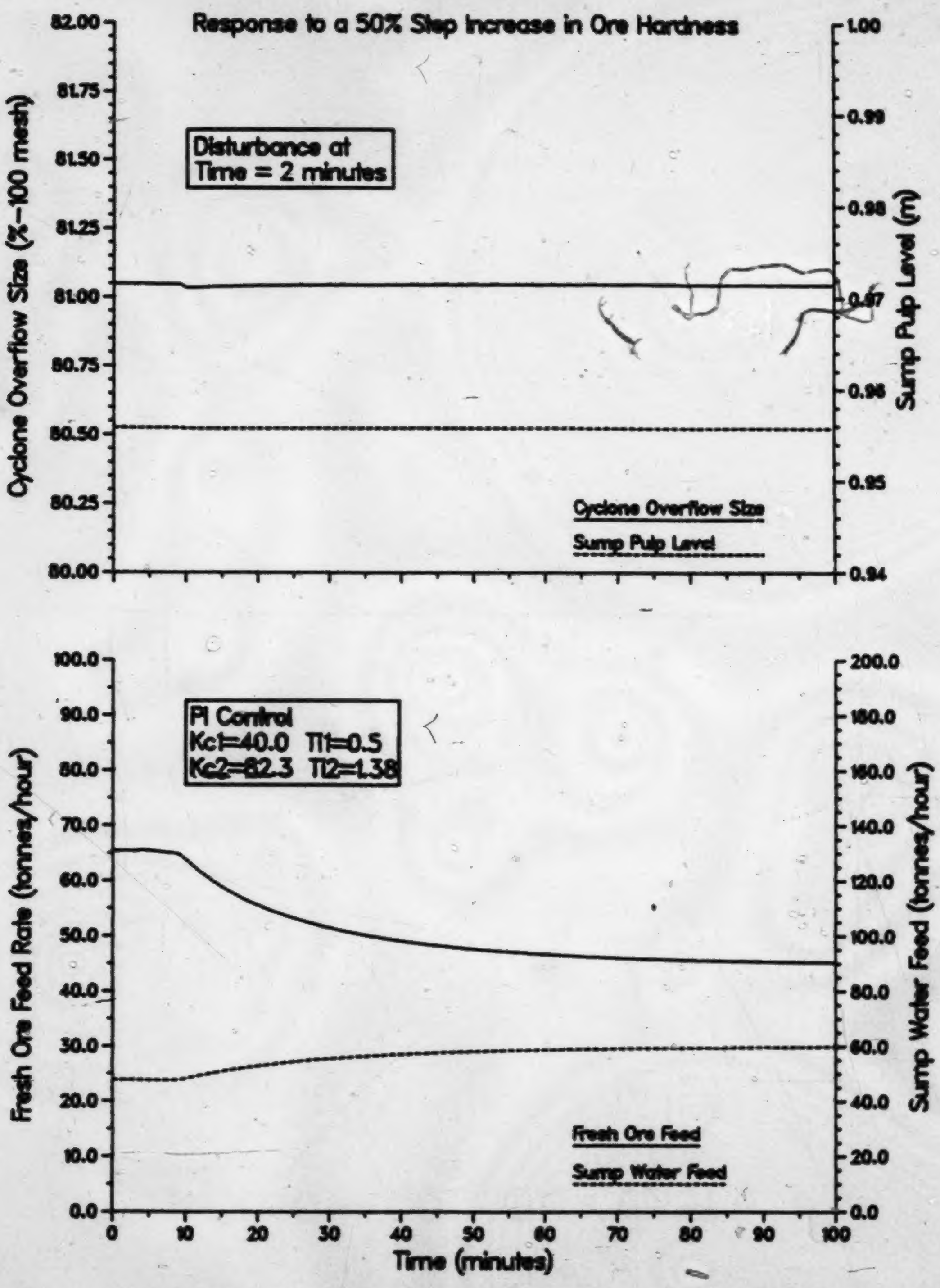


Figure 5.32: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Controller Settings

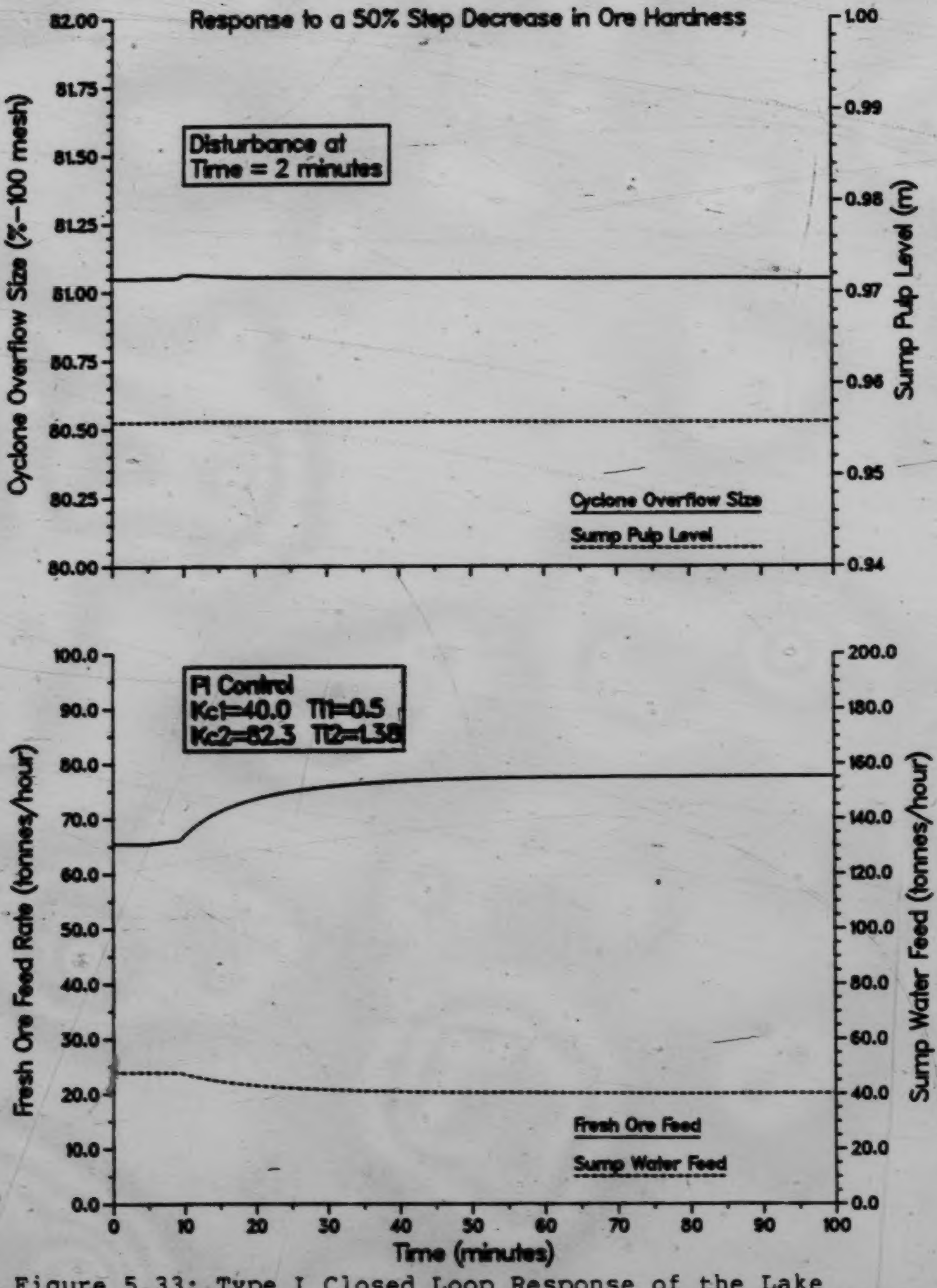


Figure 5.33: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Controller Settings

Figures 5.5 and 5.8 shows that the open loop nonlinear behavior is not evident under closed loop operation.

Depending on the downstream processing requirements, it is also possible that steady state operating conditions may need to be altered by introducing setpoint changes to the control loop(s). This is especially true of the hydrocyclone overflow size because the efficiency of the flotation circuit, the usual downstream operation, is dependent on the size of the ground particles. Thus the servo behavior of the control system is also important. Using the tuned controller constants for regulatory control, and subjecting the system to a 3% step increase in the COS setpoint produces the responses of the controlled and manipulated variables shown in Figure 5.34. It is evident from the responses of both controlled variables that the performance is poor due to the large rapid changes in the manipulated variables. To correct this, the controller constants must be detuned to slow the manipulated variables changes. This can be done in several different ways. Either the controller gains can be reduced or the integral time constants can be increased, or a combination of both. An acceptable control behavior was obtained by using a controller gain of -10.0 and an integral time constant of 2.0 minutes in loop 1, while the constants in loop 2 remained unchanged from the regulatory case as can be seen from the responses shown in Figure 5.35. The response is much slower than that shown in Figure 5.34, but

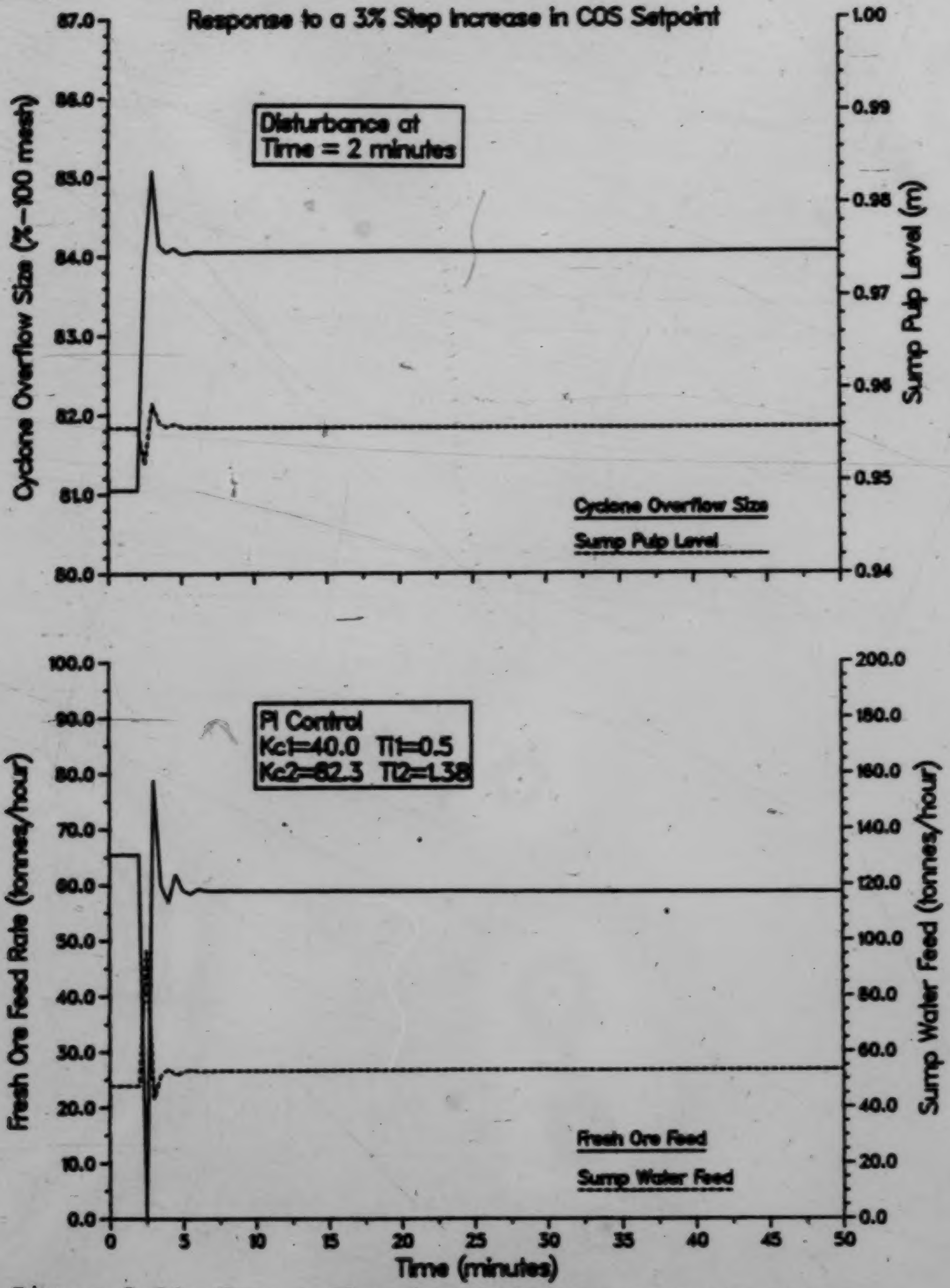


Figure 5.34: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using the Regulatory Controller Settings

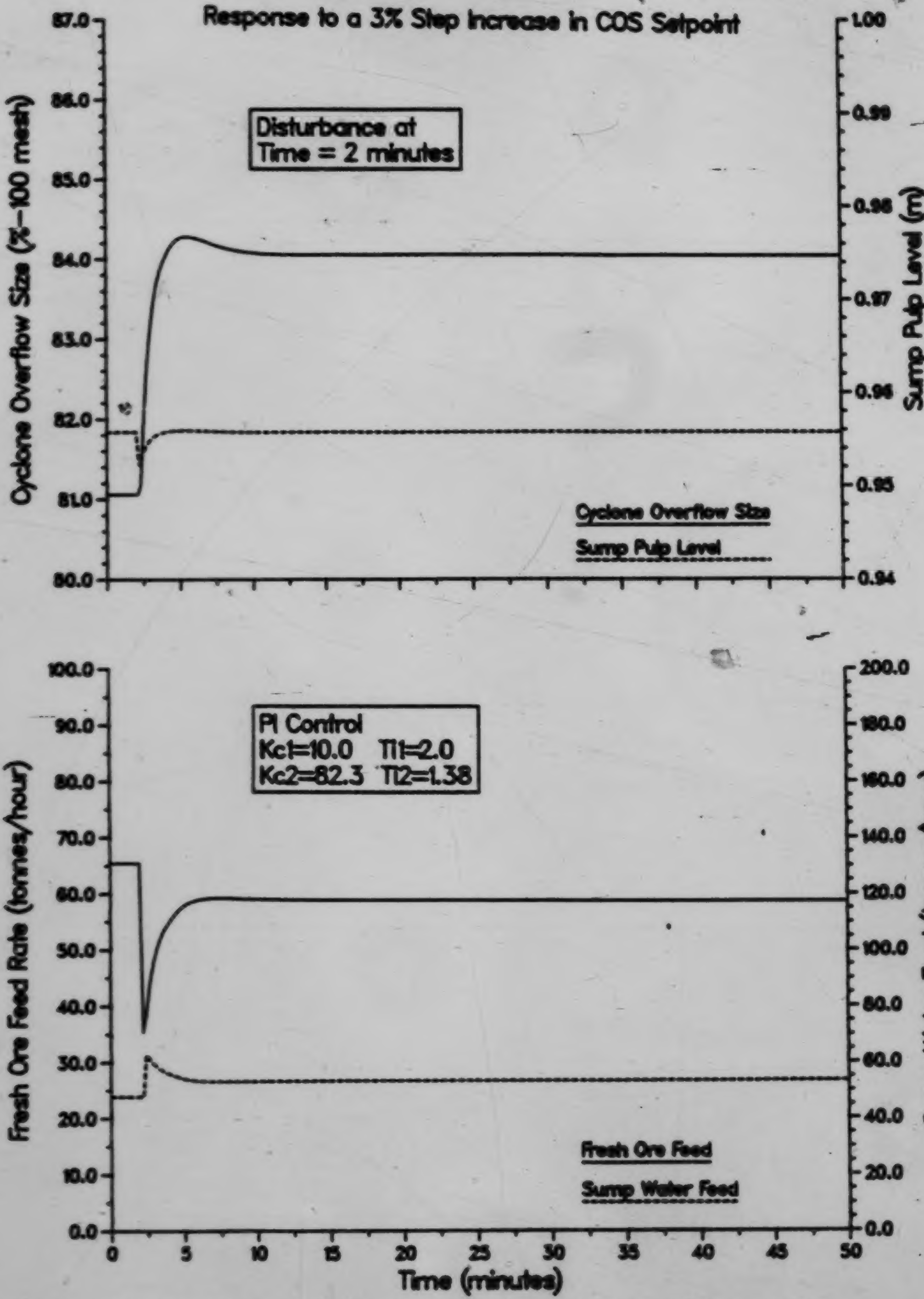


Figure 5.35: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using the Servo Controller Settings

the overshoot has been reduced and the required manipulated variable action is not as vigorous. A 3% step decrease in the COS setpoint produces the satisfactory responses shown in Figure 5.36.

The responses of the controlled and manipulated variables for ±50% step changes in ore hardness using the controller constants tuned for servo control are presented in Figures 5.37 and 5.38. It is obvious that there has been a significant deterioration in control performance compared with that shown in Figures 5.32 and 5.33.

The simulations performed for the Type I variable pairing scheme using the nonlinear circuit model are summarized in Table 5.6 along with a brief comment on the control performance in each case.

The transfer function model developed for the grinding circuit in Section 5.4 can be used in place of the nonlinear model. Using the controller constants tuned for servo control of the nonlinear model ($K_{C1} = -10.0$, $K_{C2} = 82.3$, and $\tau_{I1} = 2.0$, $\tau_{I2} = 1.38$) and subjecting the system to ±50% step changes in hardness produces the response curves shown in Figures 5.39 and 5.40. It can be seen that although these figures do not correspond exactly to the nonlinear model responses in Figures 5.37 and 5.38, the trends are very similar. The additional oscillations are due mainly to the sensitivity of the linear model to time delays.

Simulating the transfer function model under servo control conditions using the controller constants given

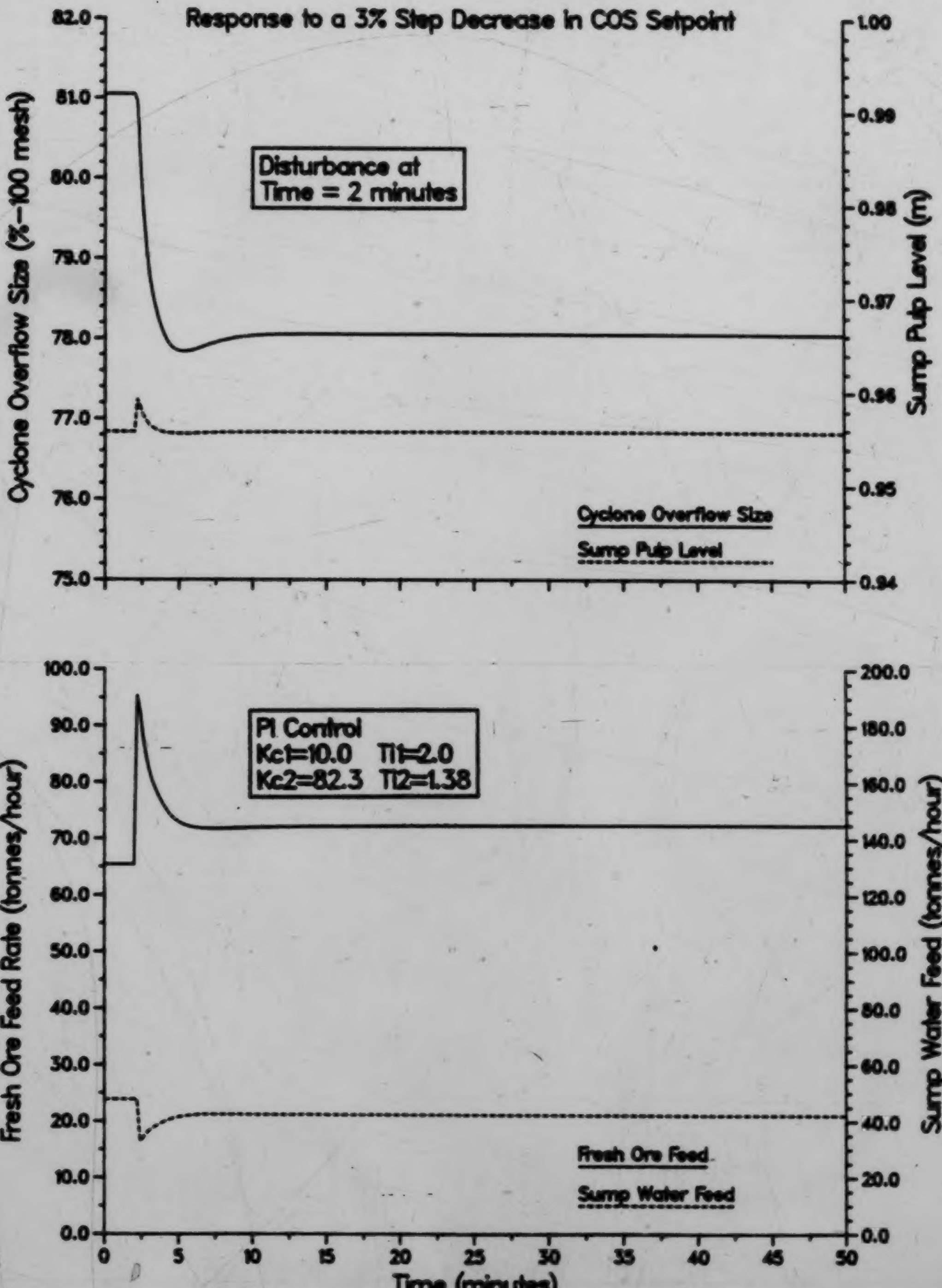


Figure 5.36: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using the Servo Controller Settings

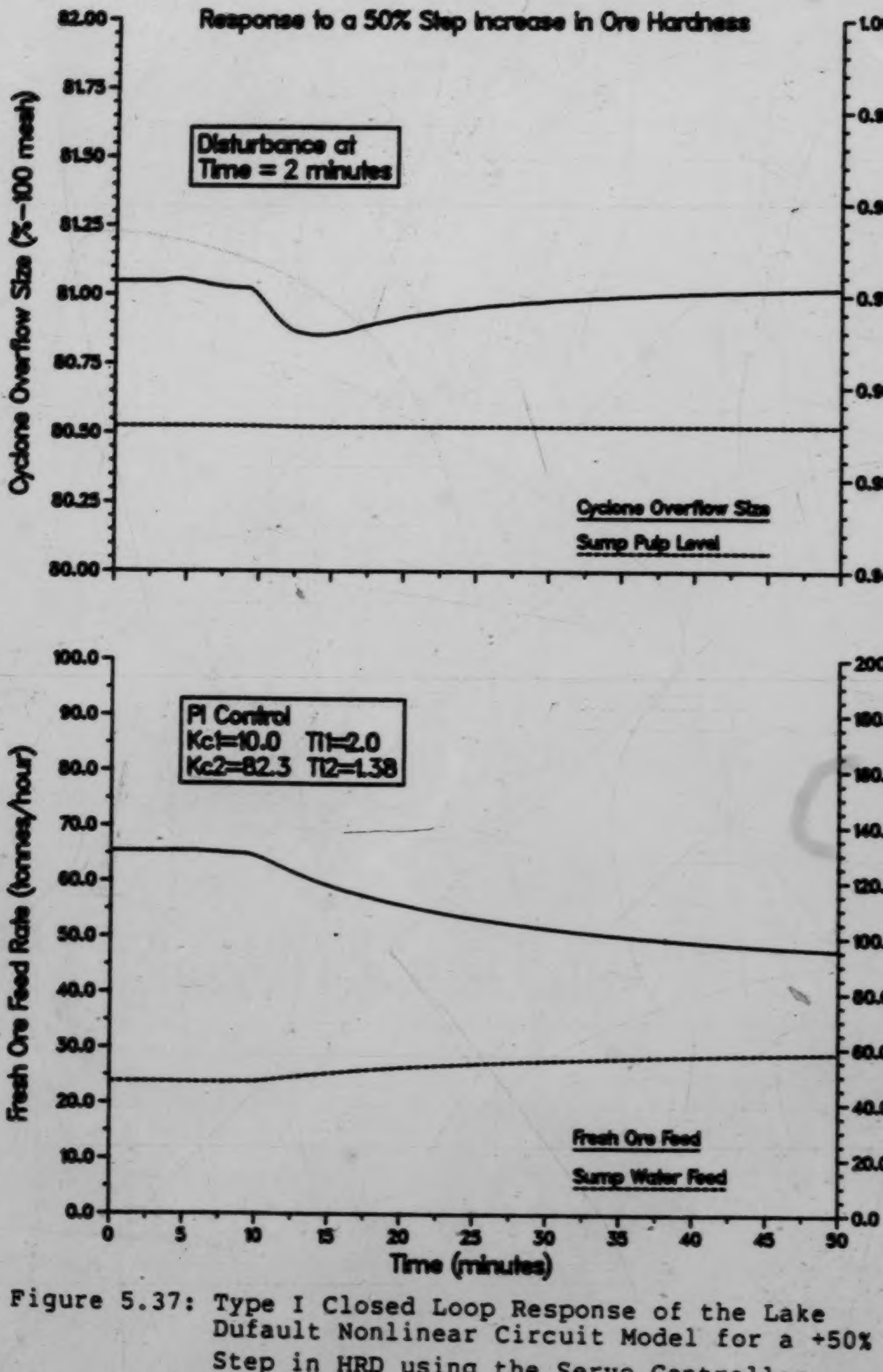


Figure 5.37: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using the Servo Controller Settings

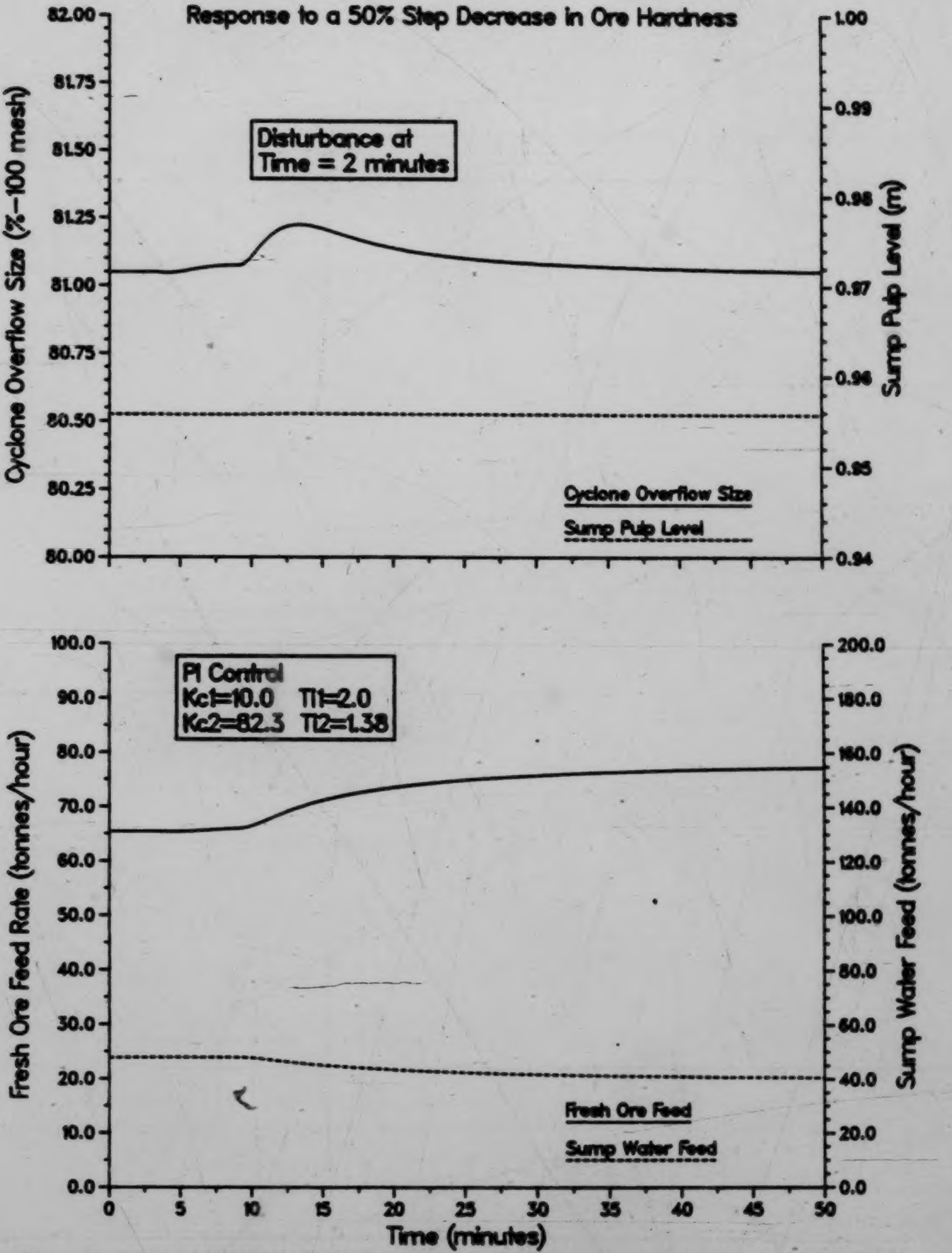


Figure 5.38: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using the Servo Controller Settings

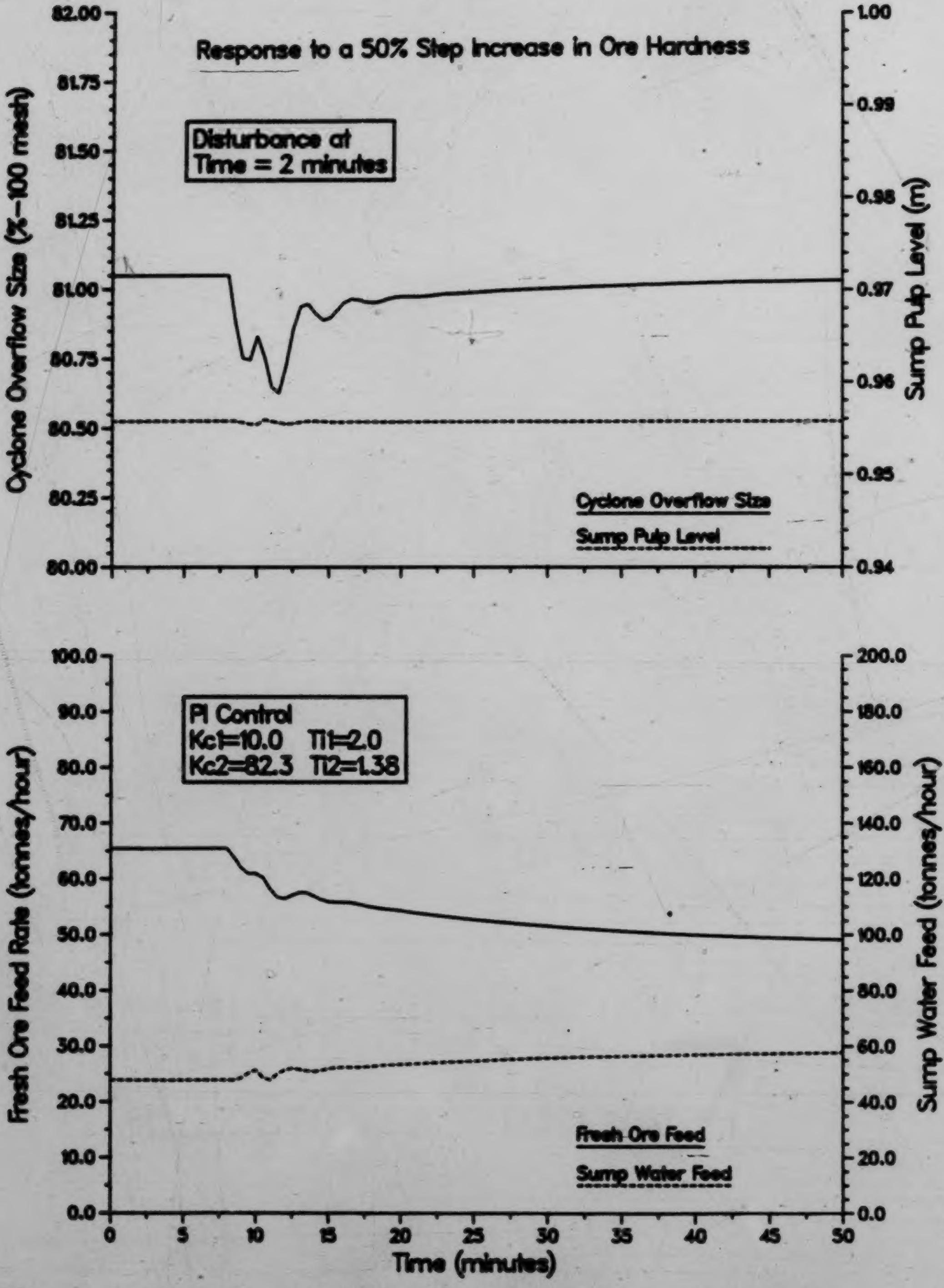


Figure 5.39: Type I Closed Loop Response of ~~the Lake~~ Dufault Linear Circuit Model for a +50% Step in HRD using the Servo Controller Settings

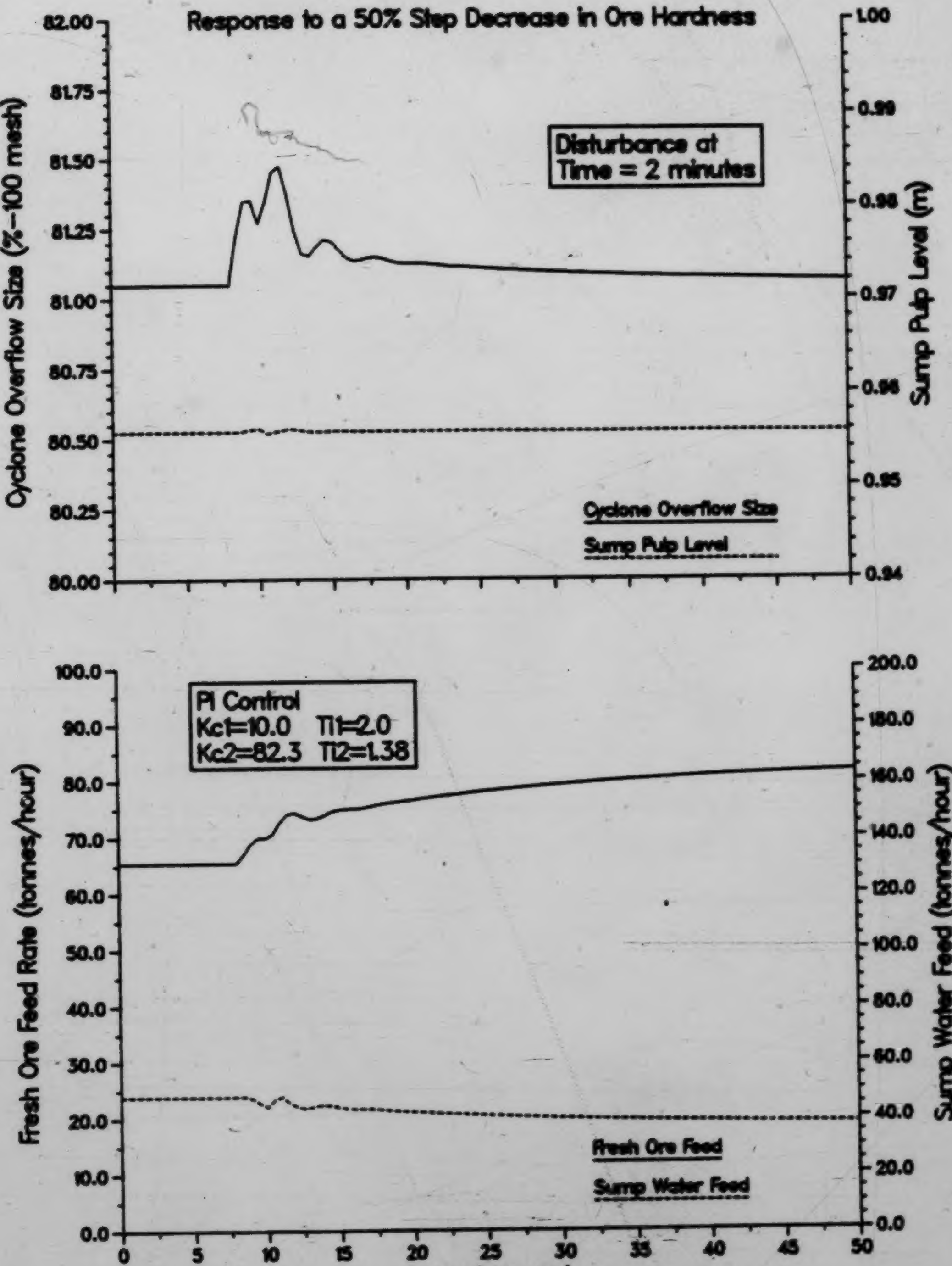


Figure 5.40: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using the Servo Controller Settings

Table 5.6: Summary of the Simulations for the Lake Dufault Grinding Circuit using the Nonlinear Model and the Type I Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.31	-2100	0.033	3600	0.33	+50% step in hardness. On-off manipulated variable action. Poor control performance.
5.32	-40.0	0.5	82.3	1.38	+50% step in hardness. Excellent control performance.
5.33	-40.0	0.5	82.3	1.38	-50% step in hardness. Excellent control performance. Nonlinear behavior absent.
5.34	-40.0	0.5	82.3	1.38	+3% step in COS setpoint. Poor control performance due to large changes in manipulated variable action.
5.35	-10.0	2.0	82.3	1.38	+3% step in COS setpoint. Excellent control performance.
5.36	-10.0	2.0	82.3	1.38	-3% step in COS setpoint. Excellent control performance.
5.37	-10.0	2.0	82.3	1.38	+50% step in hardness. Control performance degraded from Fig.5.32

Table 5.6 continued.

5.38	-10.0	2.0	82.3	1.38	-50% step in hardness. Control performance degraded from Fig.5.33

above results in the controlled and manipulated variables responses given in Figures 5.41 and 5.42. The disturbance input to the system was a ±3% step change in the COS setpoint. It can be seen from Figures 5.41 and 5.42 that the responses are again somewhat more oscillatory than the corresponding nonlinear model response curves shown in Figures 5.35 and 5.36.

A summary of the simulation results for the simulations utilizing the linear model of the grinding circuit using the Type I variable pairing scheme is given in Table 5.7.

5.6.2 Type II Control Behavior

Figure 5.43 presents a block diagram for the Type II control system configuration. It can be seen that this figure is similar to Figure 5.30 for the Type I configuration except for this control scheme, the manipulated variable used to control the hydrocyclone overflow size is the sump water feed rate (loop 1) and the manipulated variable used to control the sump pulp level is the fresh ore feed rate (loop 2). This pairing results from observing that the smaller time constants (cf. Table 5.5) are in the direct transmission paths of both loops in this configuration. This, in terms of a process transfer

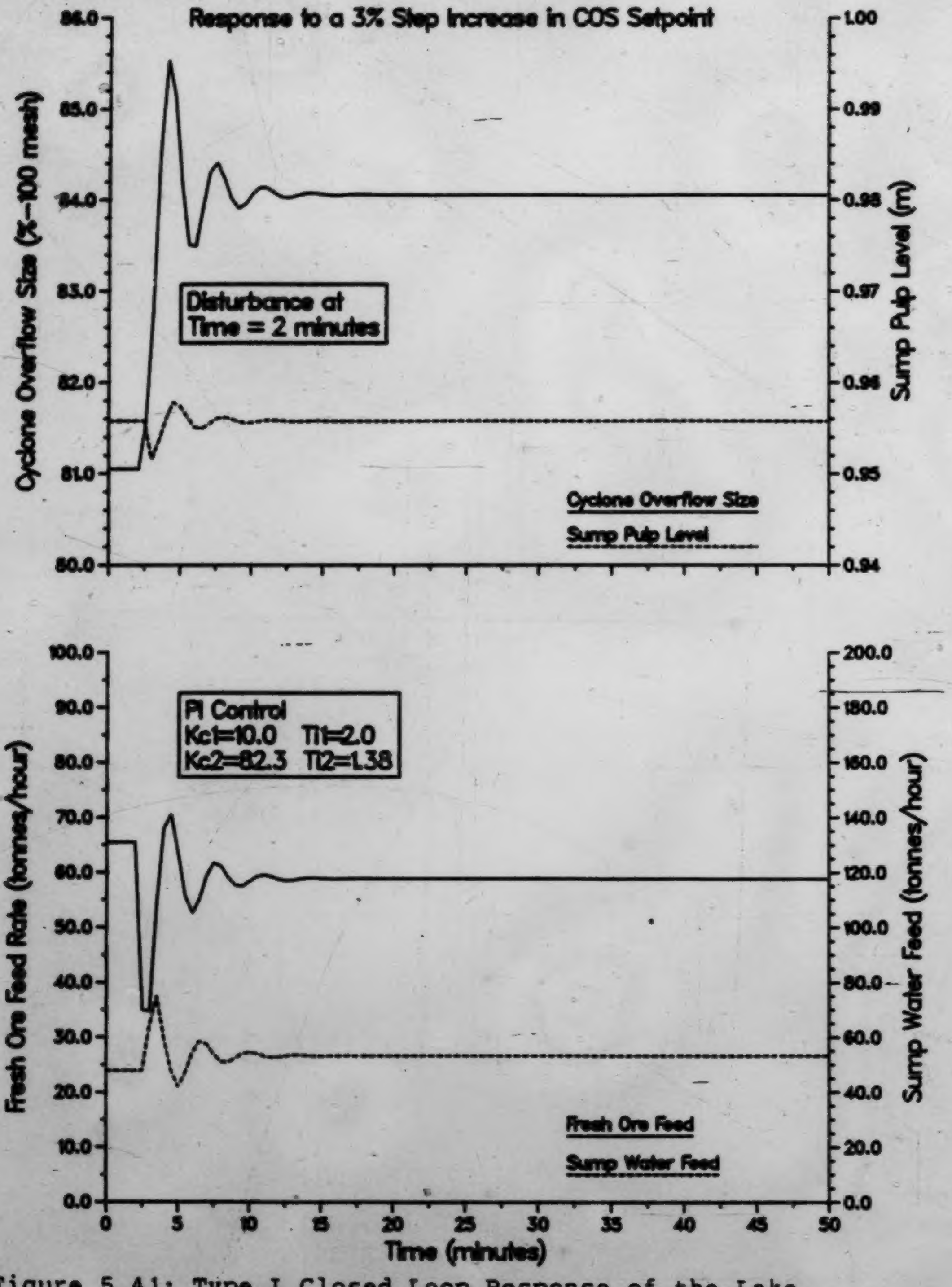


Figure 5.41: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using the Servo Controller Settings

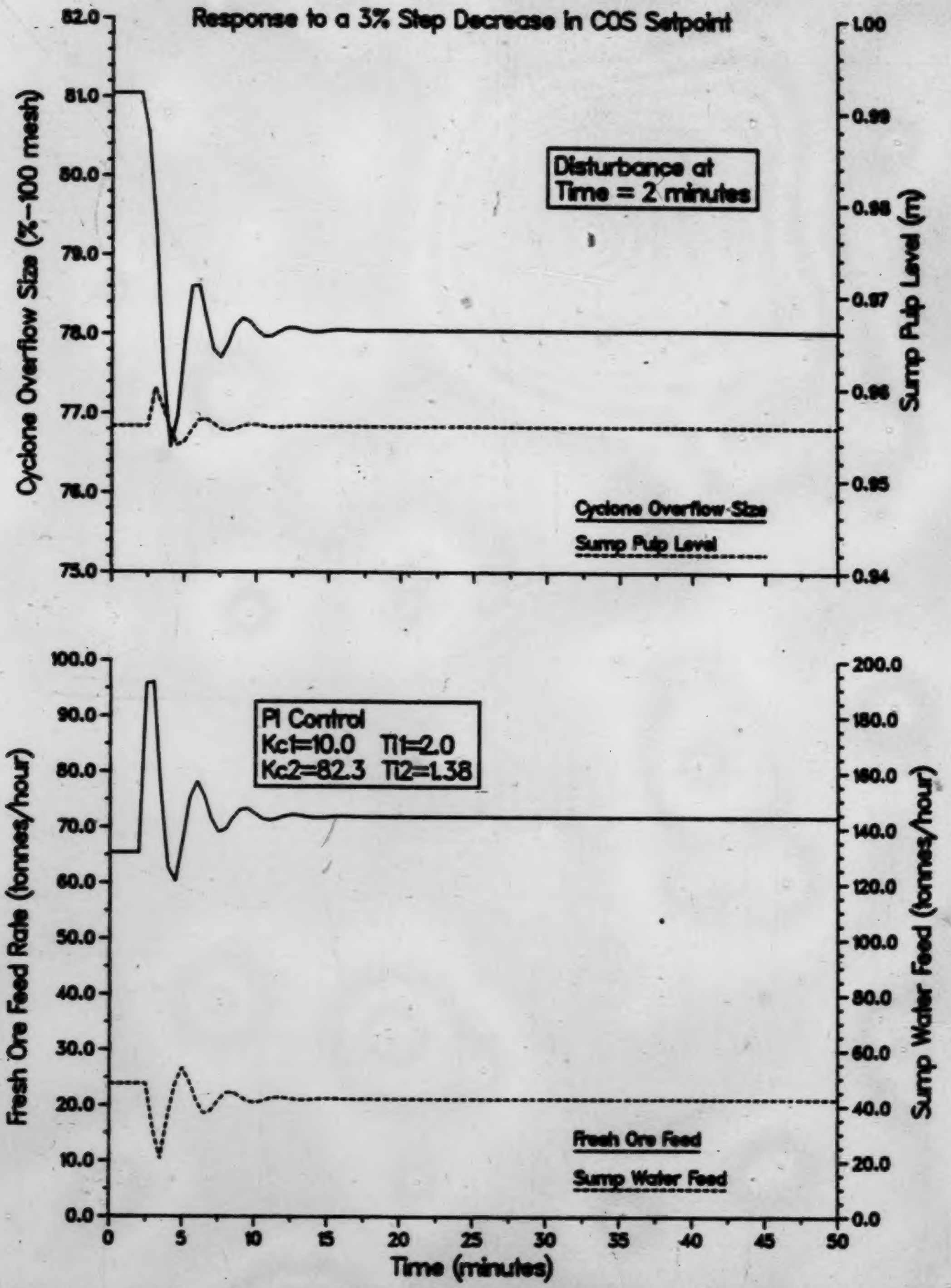


Figure 5.42: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using the Servo Controller Settings

Table 5.7: Summary of the Simulations for the Lake Dufault Grinding Circuit using the Linear Model and the Type I Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.39	-10.0	2.0	82.3	1.38	+50% step in hardness. Similar trends to nonlinear model simulation in Fig.5.37.
5.40	-10.0	2.0	82.3	1.38	-50% step in hardness. Similar trends to nonlinear model simulation in Fig.5.38.
5.41	-10.0	2.0	82.3	1.38	+3% step in COS setpoint. Increased oscillation from the nonlinear model simulation in Fig.5.35.
5.42	-10.0	2.0	82.3	1.38	+3% step in COS setpoint. Increased oscillation from the nonlinear model simulation in Fig.5.36.

function matrix, has the effect of interchanging the elements on each row from the Type I configuration, thus locating the dominant time constants on the off diagonal positions.

The initial controller tuning constants are found using the transfer function model parameters and equations (5.21) and (5.22), as was done in the last section. The resulting estimated controller constants, for loop 1, are:

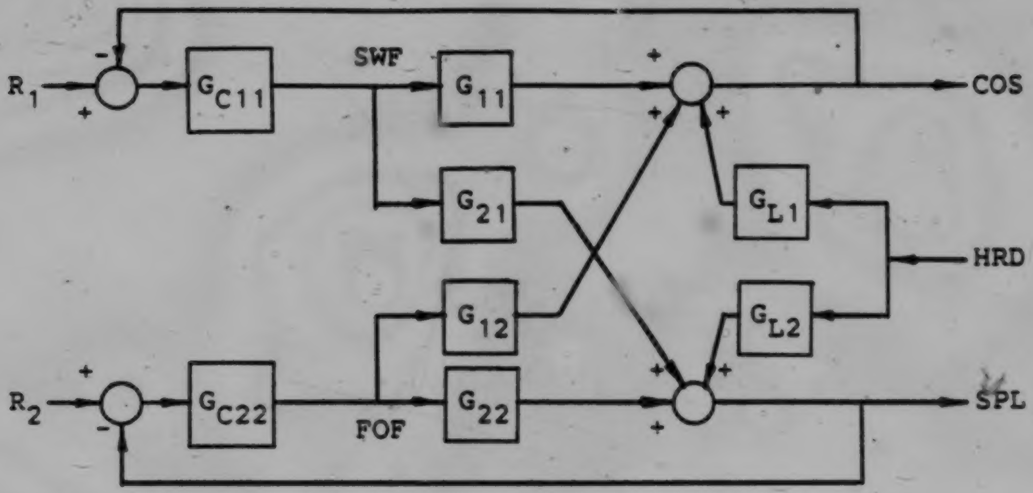

R1
R2
+
-
GC11
GC22
SWF
FOF
G11
G21
G12
G22
GL1
GL2
COS
HRD
SPL

Figure 5.43: Block Diagram Showing the Type II Control System Configuration

K_{C1} = 35.6

τ_{I1} = 0.50 minutes

and for loop 2:

K_{C2} = 700.

τ_{I2} = 2.0 minutes

Unlike the Type I configuration, both controllers require a positive gain so reverse acting controllers are necessary for both loops in this case.

Subjecting this control system to a 50% step increase in ore hardness results in very poor control, as illustrated by the responses of the controlled variables shown in Figure 5.44. The manipulated variable responses are not presented because the controllers force the variables into on-off behavior as was the case with the Type I configuration and is thought to be due to the process interaction. Reducing the gains to:

$K_{C1} = 15.0$

$K_{C2} = 41.2$

and using the integral time constants:

$\tau_{I1} = 0.5$ minutes

$\tau_{I2} = 5.0$ minutes

results in the performance shown by the responses given in

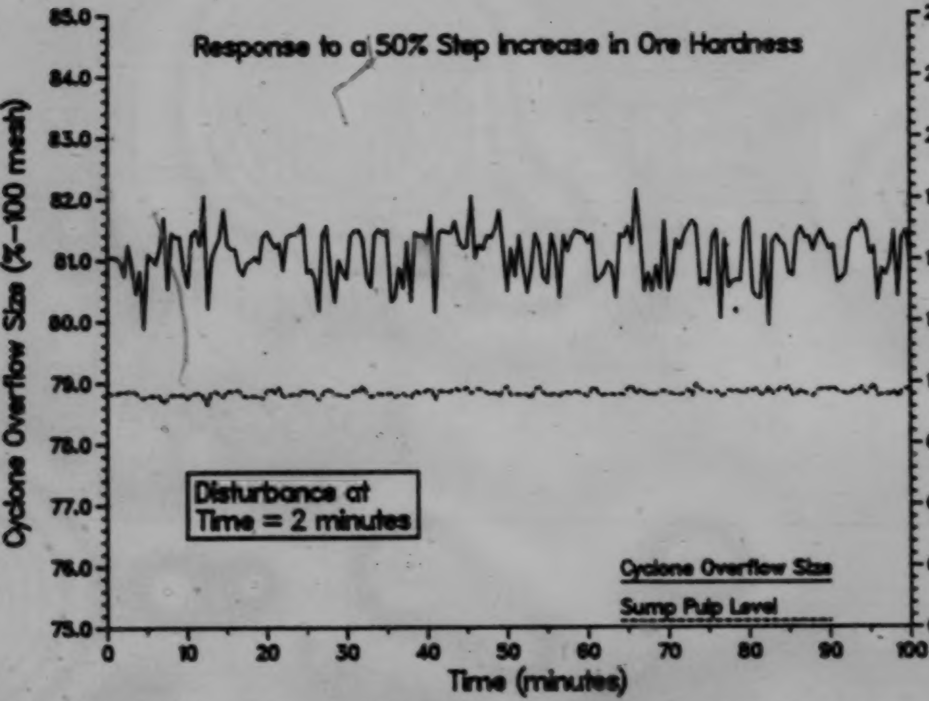


Figure 5.44: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Cohen and Coon Controller Settings

Figure 5.45 for a 50% step increase in ore hardness. The controlled and manipulated variable responses for a 50% step decrease in ore hardness are shown in Figure 5.46. These responses show a notable improvement in the overall control performance compared with that shown in Figure 5.44. It should be noted here that as with the Type I configuration, the response of the grinding circuit to changes in ore hardness is slow due to the closed circuit configuration of the grinding operation under investigation.

The responses in Figure 5.47 illustrate the effect that a +3% step change in the COS setpoint has on the grinding circuit using these tuned controllers constants. It is clear from these results that the performance of the control system is unacceptable because of the action of the manipulated variables and the large magnitude oscillations of both controlled variables. Subjecting the system to the same disturbance using detuned loop 1 controller constants of $K_{C1} = 10.0$ and $\tau_{I1} = 2.0$ minutes to slow down the response of the COS gives rise to the results shown in Figure 5.48. The corresponding response for a 3% step decrease in the setpoint of loop 1 are presented in Figure 5.49. It should be noted that it was not necessary to adjust the loop 2 controller because it is still operating in a regulatory mode. Further tuning of the controller could be carried out to improve the performance even more if desired.

Figures 5.50 and 5.51 show the system responses to

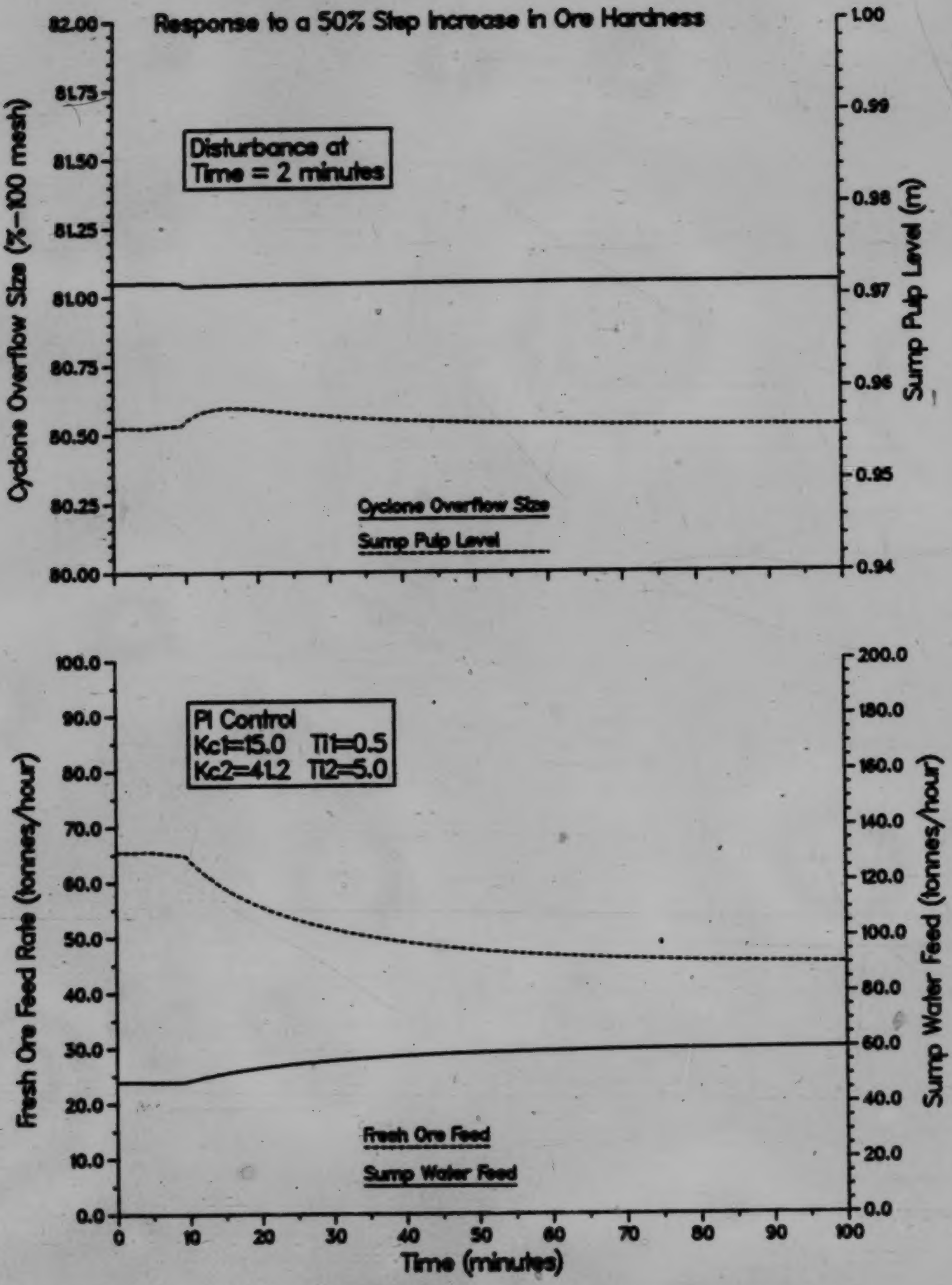


Figure 5.45: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Regulatory Controller Settings

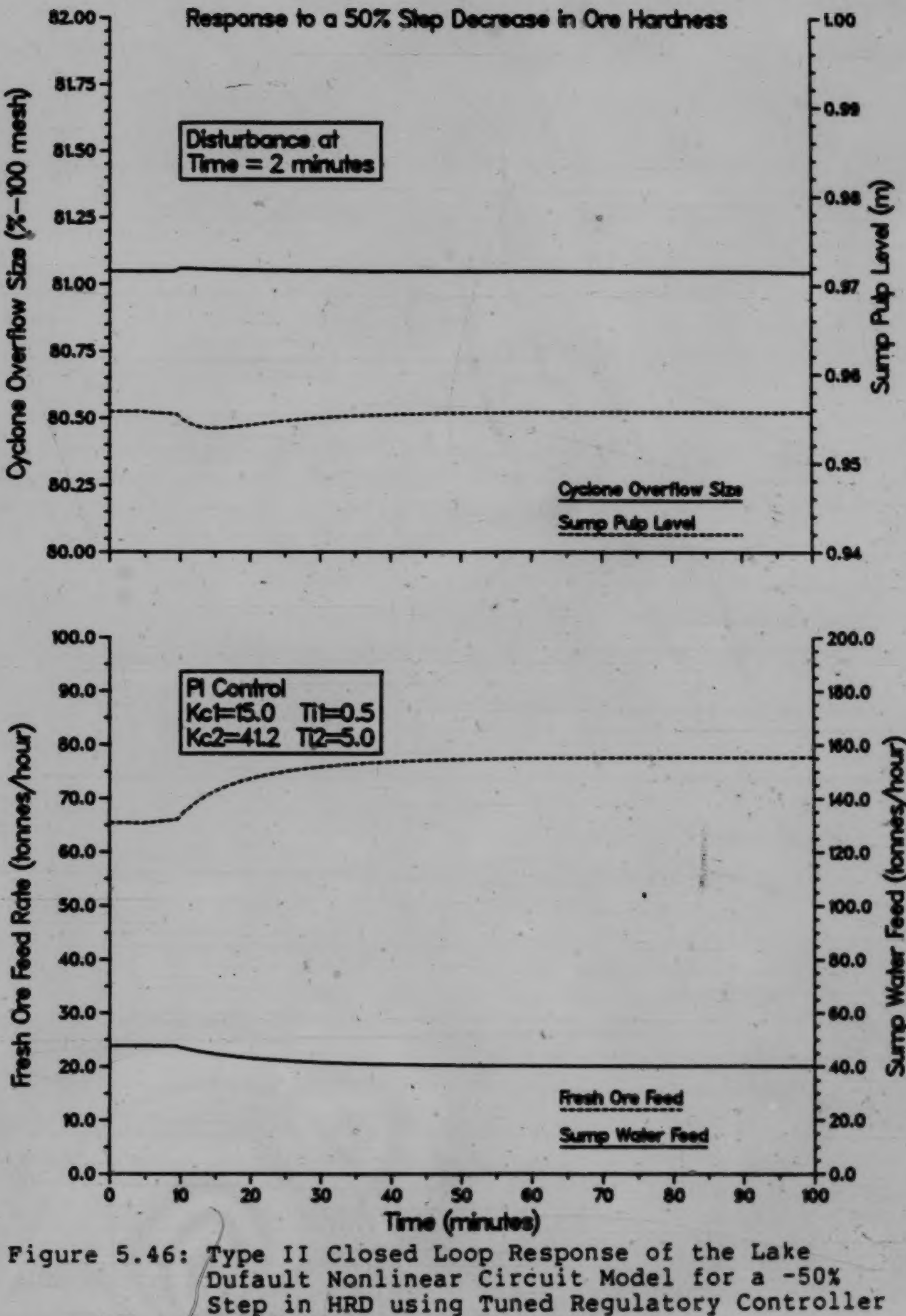


Figure 5.46: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Regulatory Controller Settings

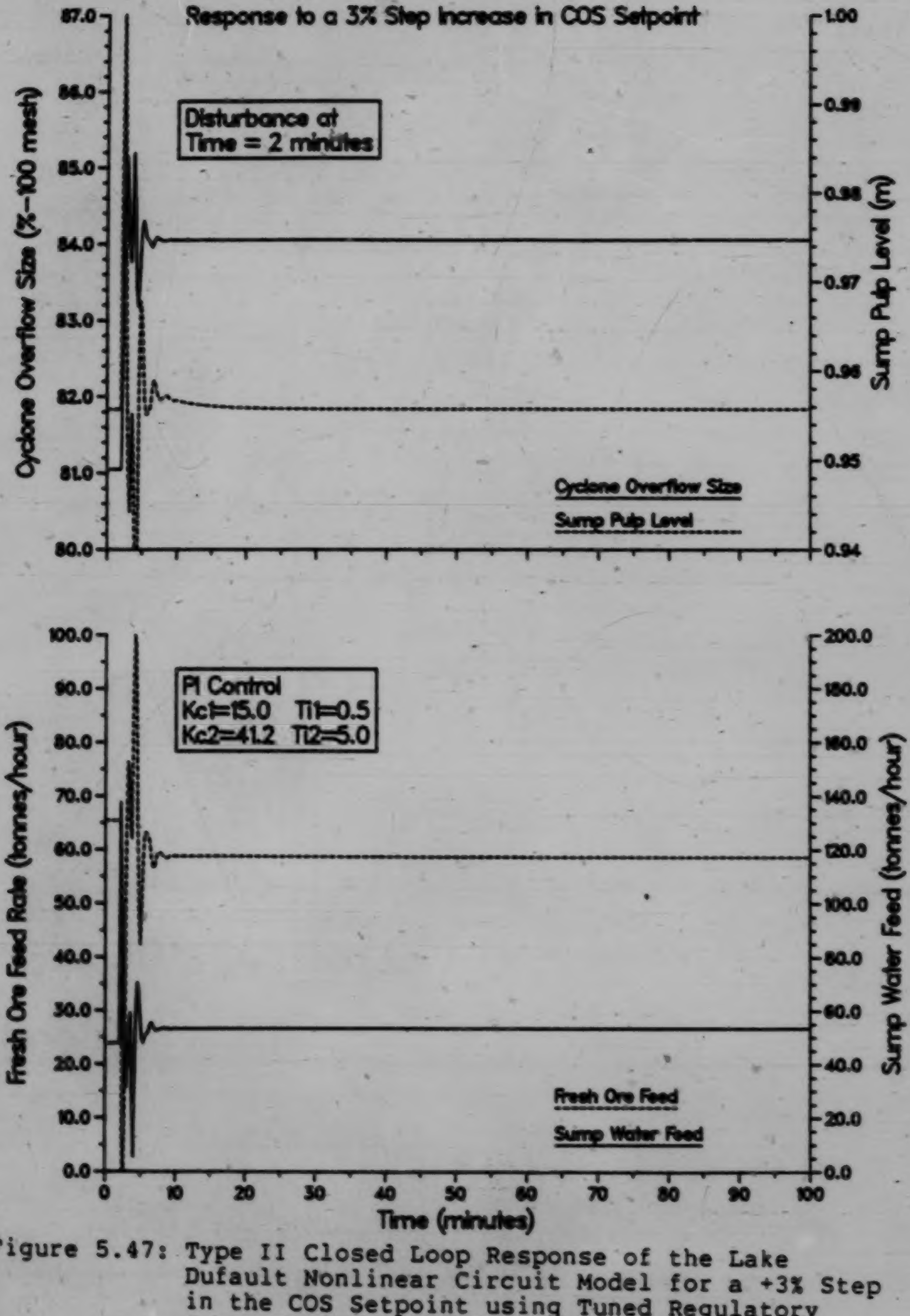


Figure 5.47: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Tuned Regulatory Controller Settings

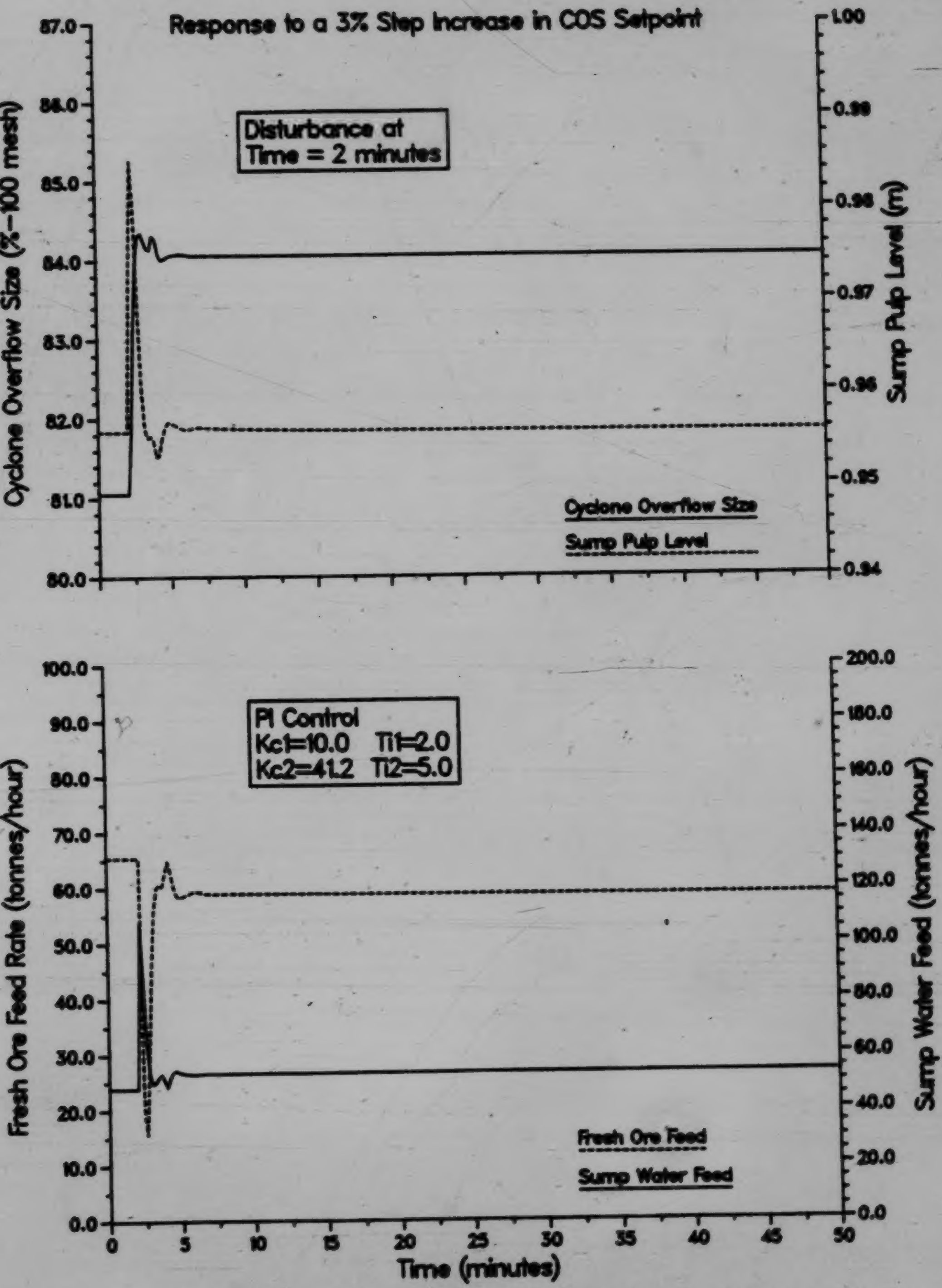


Figure 5.48: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Tuned Servo Controller Settings

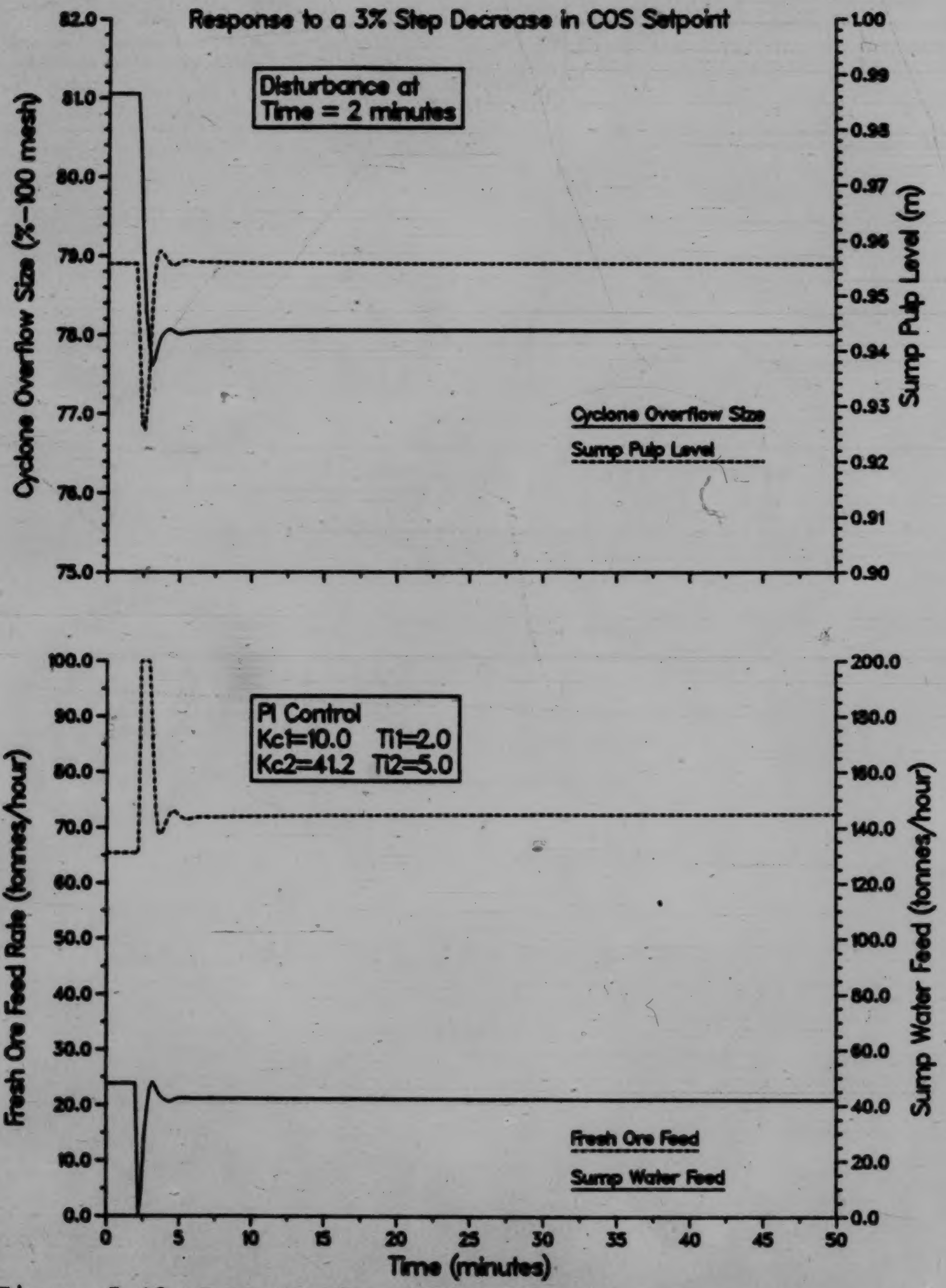


Figure 5.49: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Tuned Servo Controller Settings

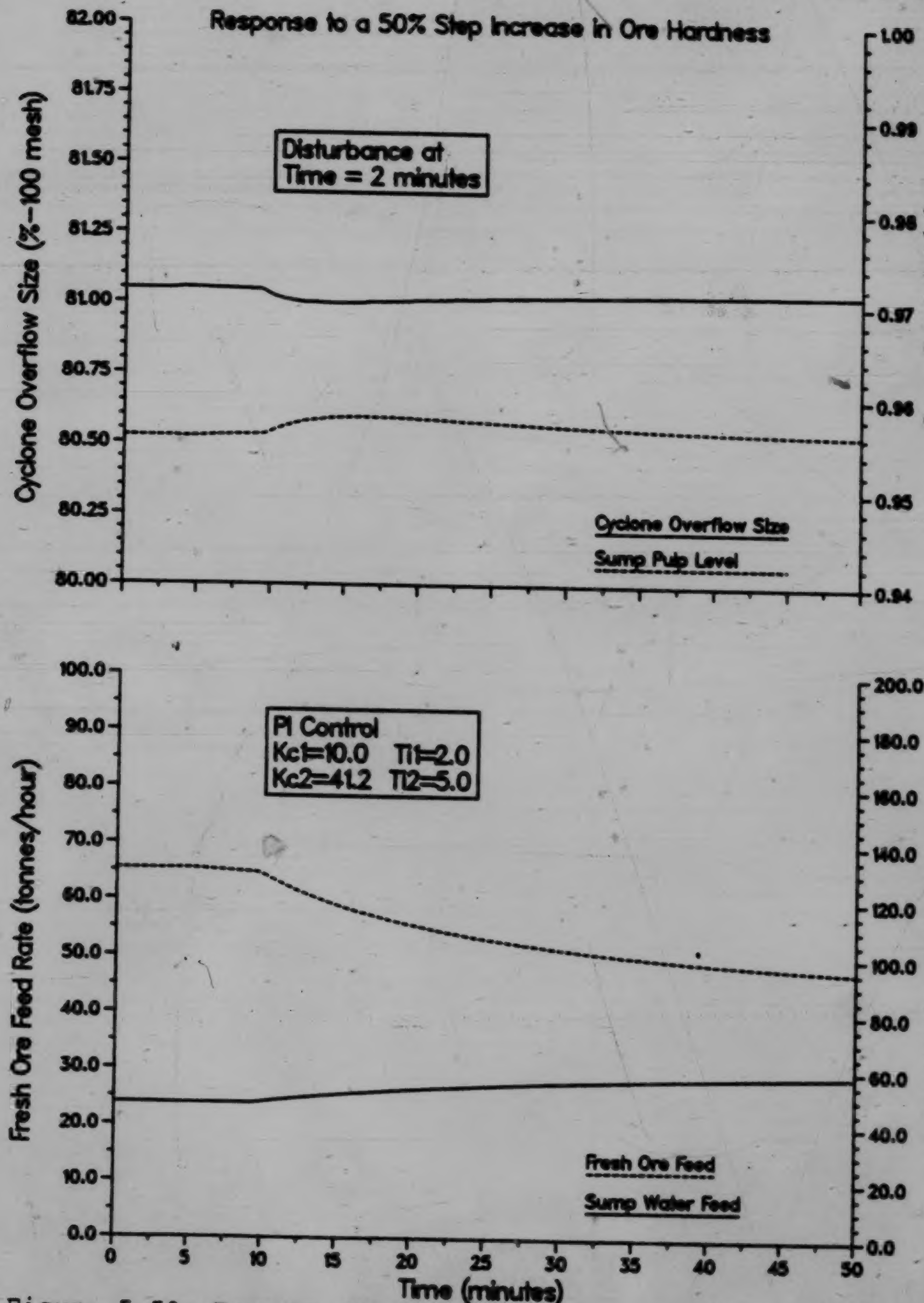


Figure 5.50: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Tuned Servo Controller Settings

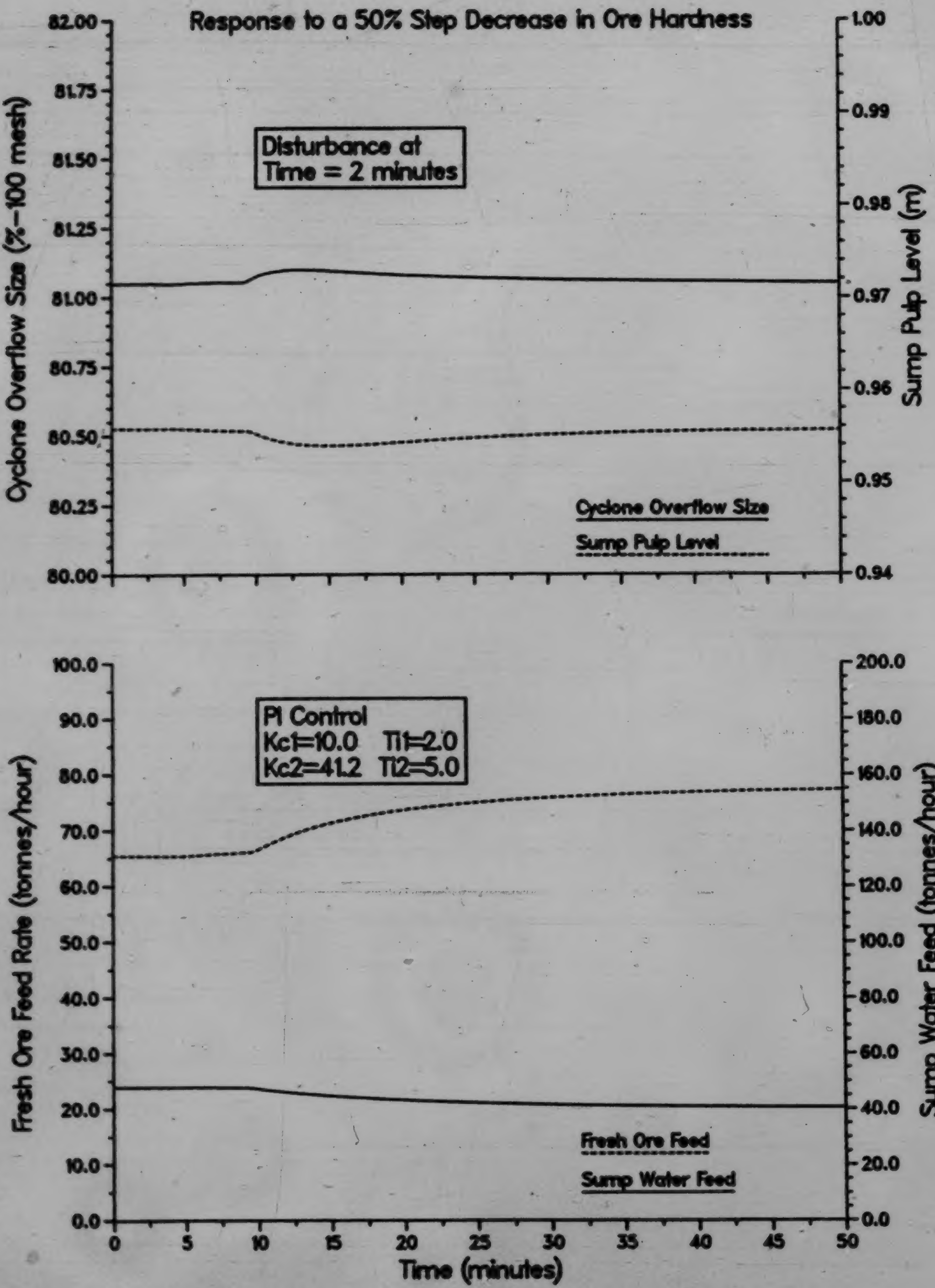


Figure 5.51: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Tuned Servo Controller Settings

positive and negative hardness changes using the controller constants tuned for satisfactory servo control. It can be seen that there has been a degradation in the control performance of the system with respect to the rejection of hardness disturbances by comparing these responses with those in Figures 5.45 and 5.46.

The simulations performed for the Type II variable pairing scheme using the nonlinear circuit model are summarized in Table 5.8 along with a brief comment on the control performance in each case.

The transfer function model developed for the grinding circuit in Section 5.4 can be used in place of the nonlinear model. Using the controller constants, $K_{C1} = 3.0$, $K_{C2} = 20.6$, and $\tau_{I1} = 2.0$, $\tau_{I2} = 5.0$, and subjecting the transfer function model to ±50% step changes in the ore hardness, produces the response curves shown in Figures 5.52 and 5.53. It should be noted that, unlike the Type I case, the controller gains had to be reduced from those used when simulating the nonlinear model under the same conditions as used here. This is due to the presence of larger time delays and longer time constants of the direct transmission transfer functions when using the Type II control scheme rather than the Type I scheme.

Simulating the transfer function model under servo control conditions using the controller constants given above results in the controlled and manipulated variables responses given in Figures 5.54 and 5.55. The disturbance

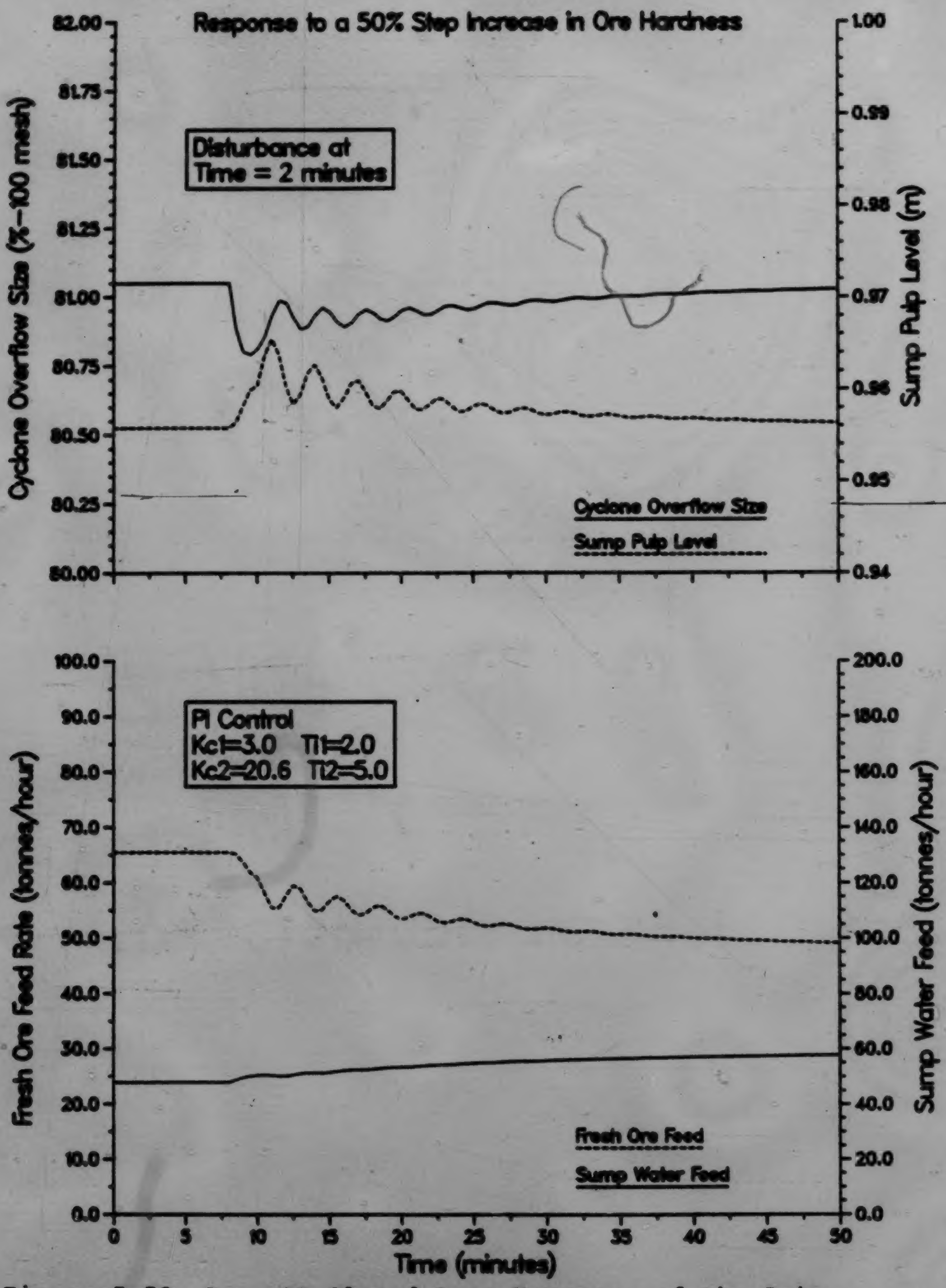


Figure 5.52: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Tuned Servo Controller Settings

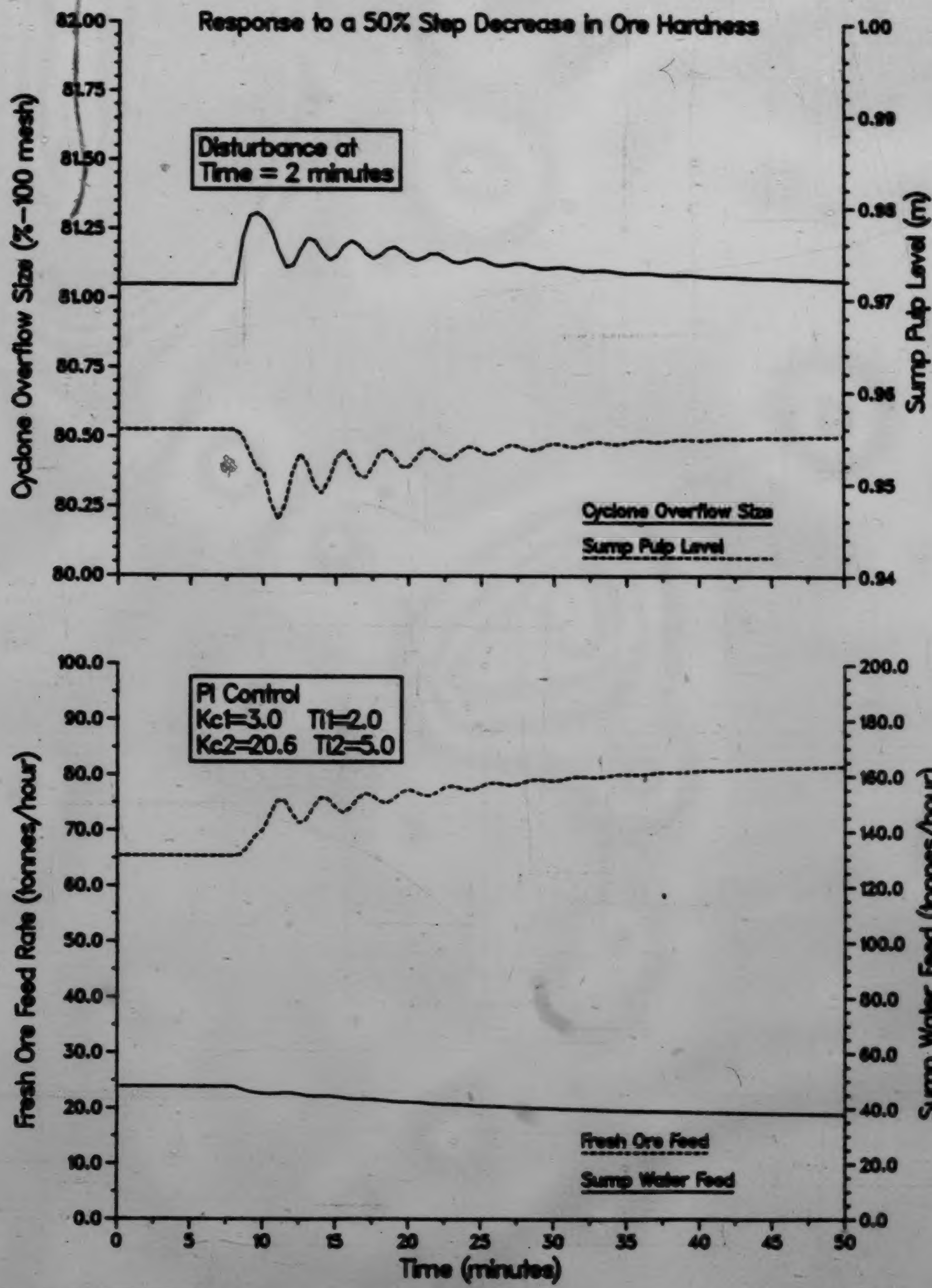


Figure 5.53: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Tuned Servo Controller Settings

Table 5.8: Summary of the Simulations for the Lake Dufault Grinding Circuit using the Nonlinear Model and the Type II Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.44	35.6	0.5	700	2.0	+50% step in hardness. On-off manipulated variable action. Poor control performance.
5.45	15.0	0.5	41.2	5.0	+50% step in hardness. Excellent control performance.
5.46	15.0	0.5	41.2	5.0	-50% step in hardness. Excellent control performance. Nonlinear behavior absent.
5.47	15.0	0.5	41.2	5.0	+3% step in COS setpoint. Poor control performance due to large changes in manipulated variable action.
5.48	10.0	2.0	41.2	5.0	+3% step in COS setpoint. Good control performance.
5.49	10.0	2.0	41.2	5.0	-3% step in COS setpoint. Good control performance.
5.50	10.0	2.0	41.2	5.0	+50% step in hardness. Control performance degraded from Fig.5.45

Table 5.8 continued.

5.51	10.0	2.0	41.2	5.0	-50% step in hardness. Control performance degraded from Fig.5.46

input to the system was a ±3% step change in the COS setpoint. It should be noted that these settings have been detuned from the servo controller constants used with the nonlinear model due to the sensitivity of the transfer function model to time delays.

A summary of the simulation results for the simulations utilizing the linear model of the grinding circuit using the Type II variable pairing scheme is given in Table 5.9.

5.7 Delay Compensator Design

The first step in attempting to improve the performance of the Type I and Type II control system configurations is to compensate for the time delays in the process. The multivariable time delay compensation technique of Ogunnaike and Ray outlined in Chapter 2 (cf. Section 2.6.4) will be used. This technique was chosen because of its simialrity to the SISO Smith predictor.

The design of the time delay compensator follows from equations (2.89) through (2.92) and the transfer function model derived in Section 5.4. The undelayed transfer function model is given in Table 5.10. Thus, for the Type I control system configuration, the time delay compensator is given by:

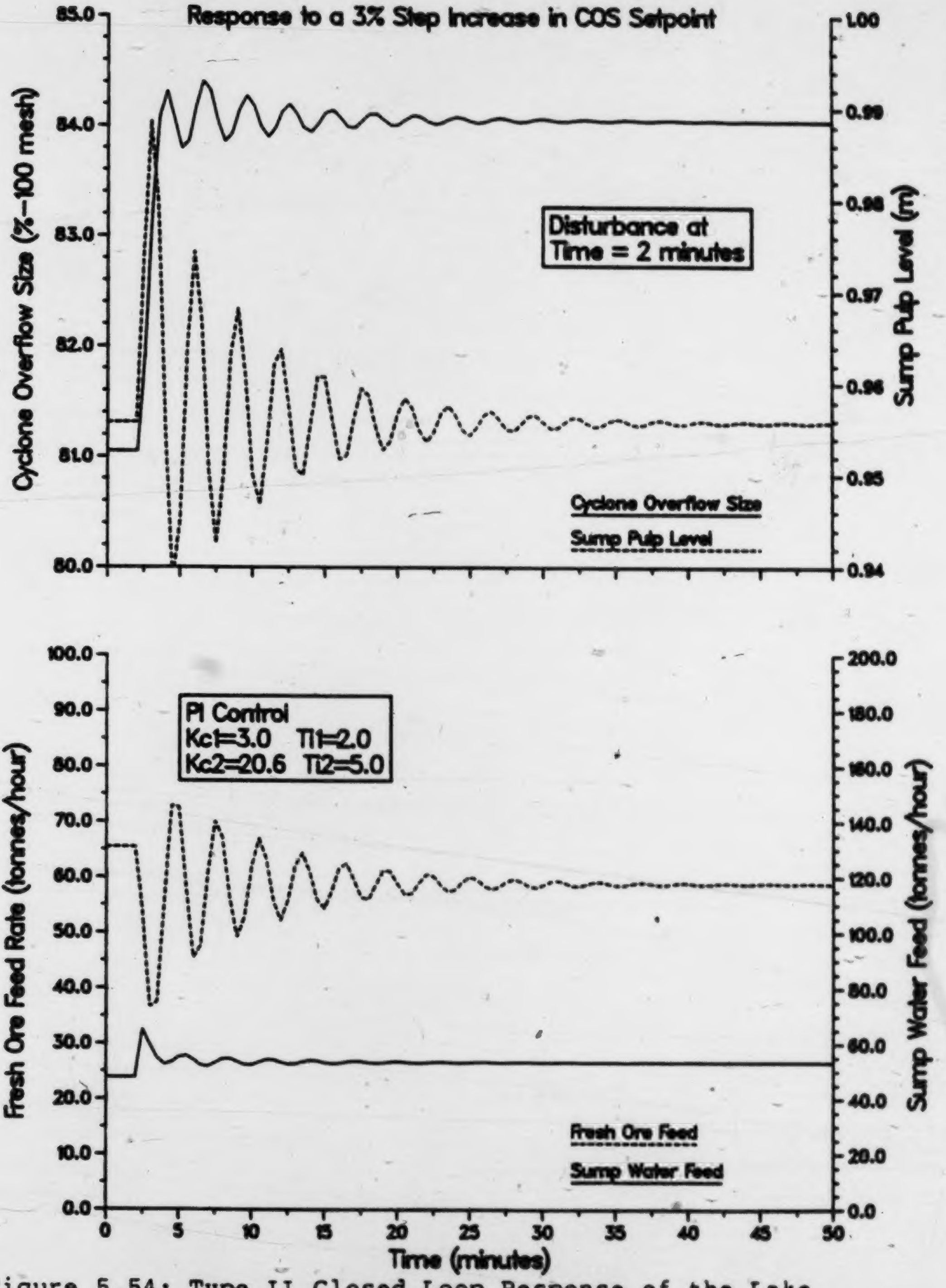


Figure 5.54: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Tuned Servo Controller Settings

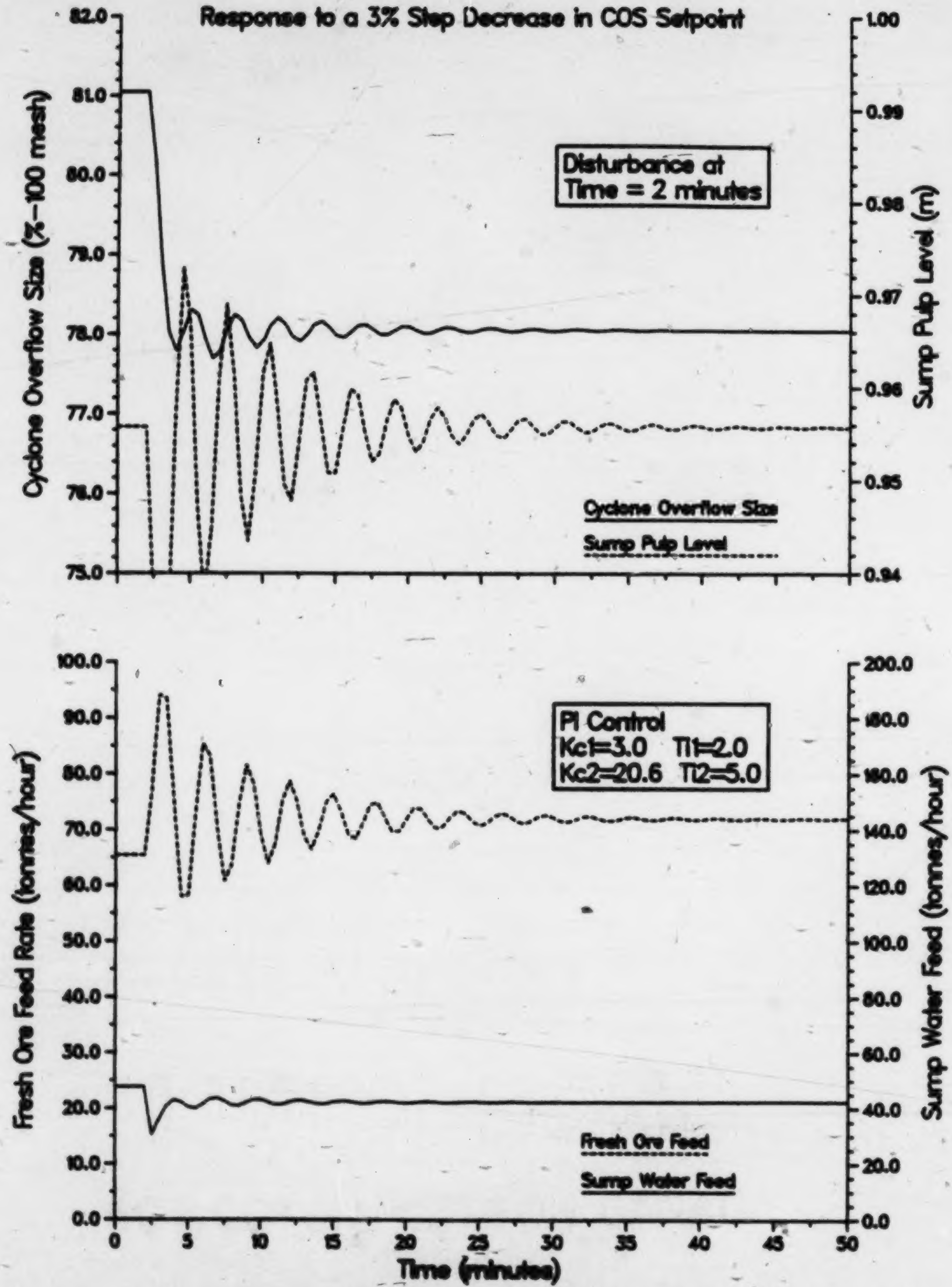


Figure 5.55: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Tuned Servo Controller Settings

Table 5.9: Summary of the Simulations for the Lake Dufault Grinding Circuit using the Linear Model and the Type II Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.52	3.0	2.0	20.6	5.0	+50% step in hardness. Similar trends to nonlinear model simulation in Fig.5.50.
5.53	3.0	2.0	20.6	5.0	-50% step in hardness. Similar trends to nonlinear model simulation in Fig.5.51.
5.54	3.0	2.0	20.6	5.0	+3% step in COS setpoint. Increased oscillation from the nonlinear model simulation in Fig.5.48.
5.55	3.0	2.0	20.6	5.0	+3% step in COS setpoint. Increased oscillation from the nonlinear model simulation in Fig.5.49.

$$G_k = \begin{bmatrix} 0 & \dfrac{0.337(1-e^{-0.18s})}{2.32s+1} \\ \dfrac{0.199(1-e^{-0.42s})}{80.1s+1} & \dfrac{0.161(1-e^{-1.62s})}{77.6s+1} \end{bmatrix} \qquad (5.23)$$

while for the Type II control configuration, the time delay compensator is:

Table 5.10: A Two by Two Transfer Function Model of the Lake Dufault Mineral Grinding Circuit Without Time Delays

Manipulated (Input) Variable	Controlled (Output) Variable	
	COS	SPL
FOF	$\frac{-0.179}{4.18s+1}$	$\frac{0.161}{77.6s+1}$
SWF	$\frac{0.337}{2.32s+1}$	$\frac{0.199}{80.1s+1}$

$$G_k = \begin{bmatrix} \dfrac{0.337(1-e^{-0.18s})}{2.32s+1} & 0 \\ \dfrac{0.161(1-e^{-1.62s})}{77.6s+1} & \dfrac{0.199(1-e^{-0.42s})}{80.1s+1} \end{bmatrix} \qquad (5.24)$$

5.8 Time Delay Compensated Behavior

The manipulated and controlled variable responses of the Lake Dufault grinding circuit using the nonlinear model as the actual plant, under Type I control for a 50% step increase and decrease in the ore hardness (HRD) when using the time delay compensator given in equation (5.23) are presented in Figures 5.56 and 5.57. It is clear from the responses shown in these figures that the performance of the system shows little improvement in the response of the COS controlled variable over the uncompensated sytem responses

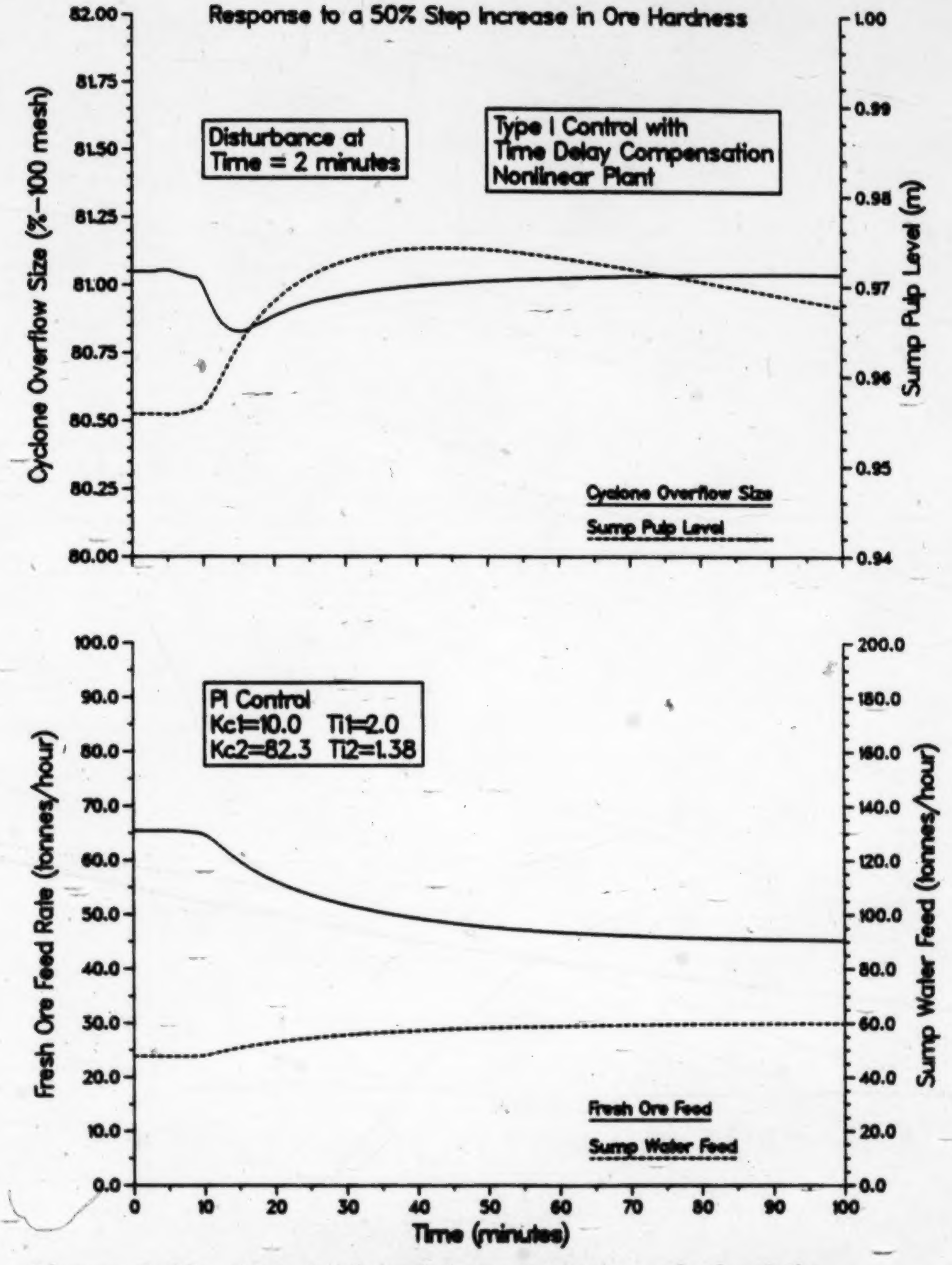


Figure 5.56: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Time Delay Compensation

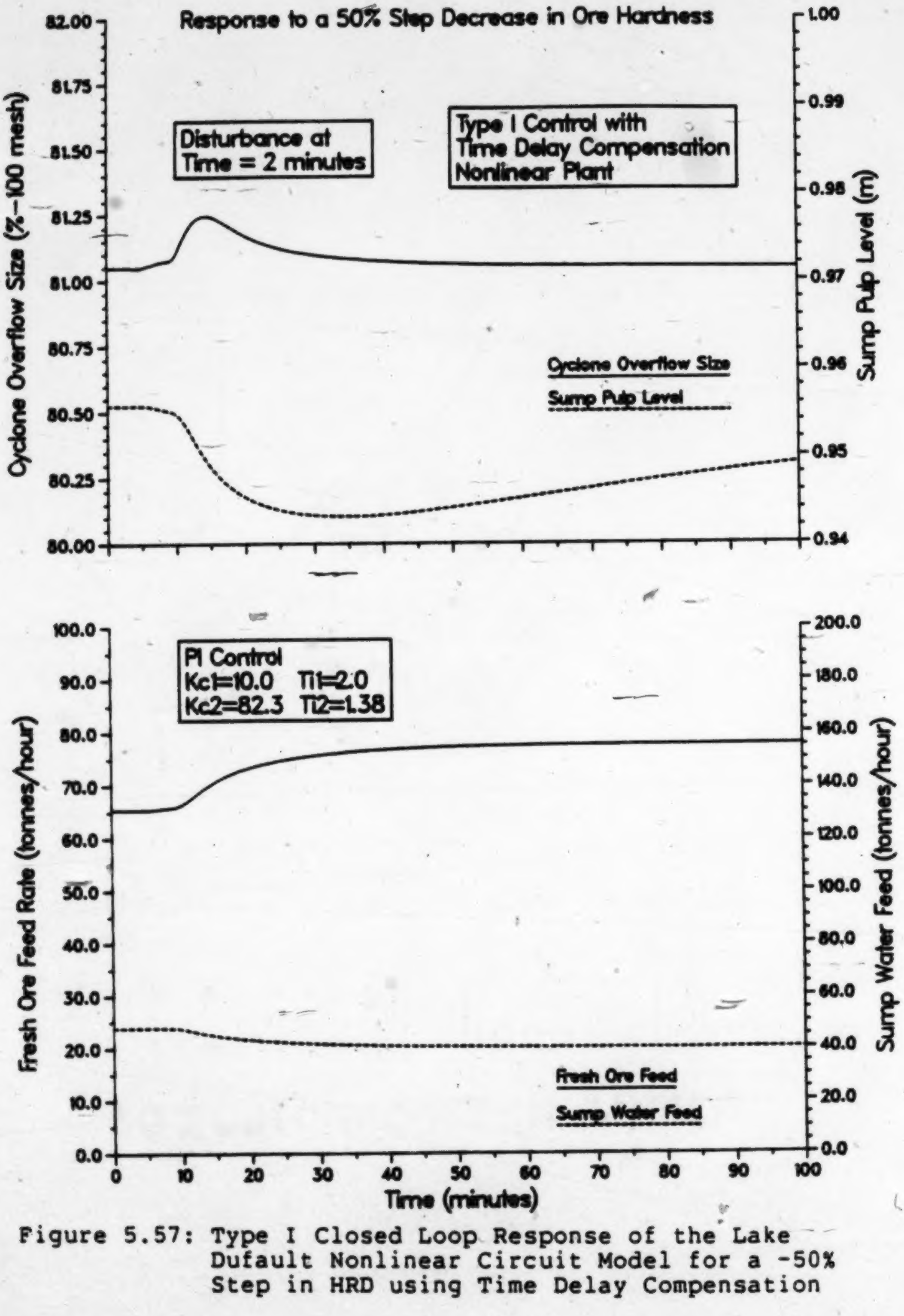


Figure 5.57: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Time Delay Compensation

shown in Figures 5.37 and 5.38. In fact, the performance of loop 2 (SPL) has been degraded. This degradation in performance from the original Type I configuration simulations is likely due to the time delay sensitivity of the linear model used in the time delay compensator as well as the existance of model mismatch between the linear and nonlinear grinding circuit models. It should be noted here that although the response of the sump pulp level controlled variable in Figures 5.56 and 5.57 appears to be extremely large, the maximum deviation from the setpoint is less than two centimetres, which in a level control system may be satisfactory.

The model mismatch is demonstrated by using the linear model of the grinding circuit instead of the nonlinear model as the actual plant. This produces the results illustrated in Figures 5.58 and 5.59. A comparison of these responses with those given in Figures 5.56 and 5.57 clearly shows that the largest difference in responses occurs for loop 1, indicating that this is the source of model mismatch.

Control performance and the response of the manipulated variables of the time delay compensated system obtained using the nonlinear circuit for ±3% step changes in the hydrocyclone overflow size (COS) is presented in Figures 5.60 and 5.61. If the circuit is modelled with the linear model, the responses that result are shown in Figures 5.62 and 5.63. The controller constants used for both the nonlinear and linear model simulations were the same as

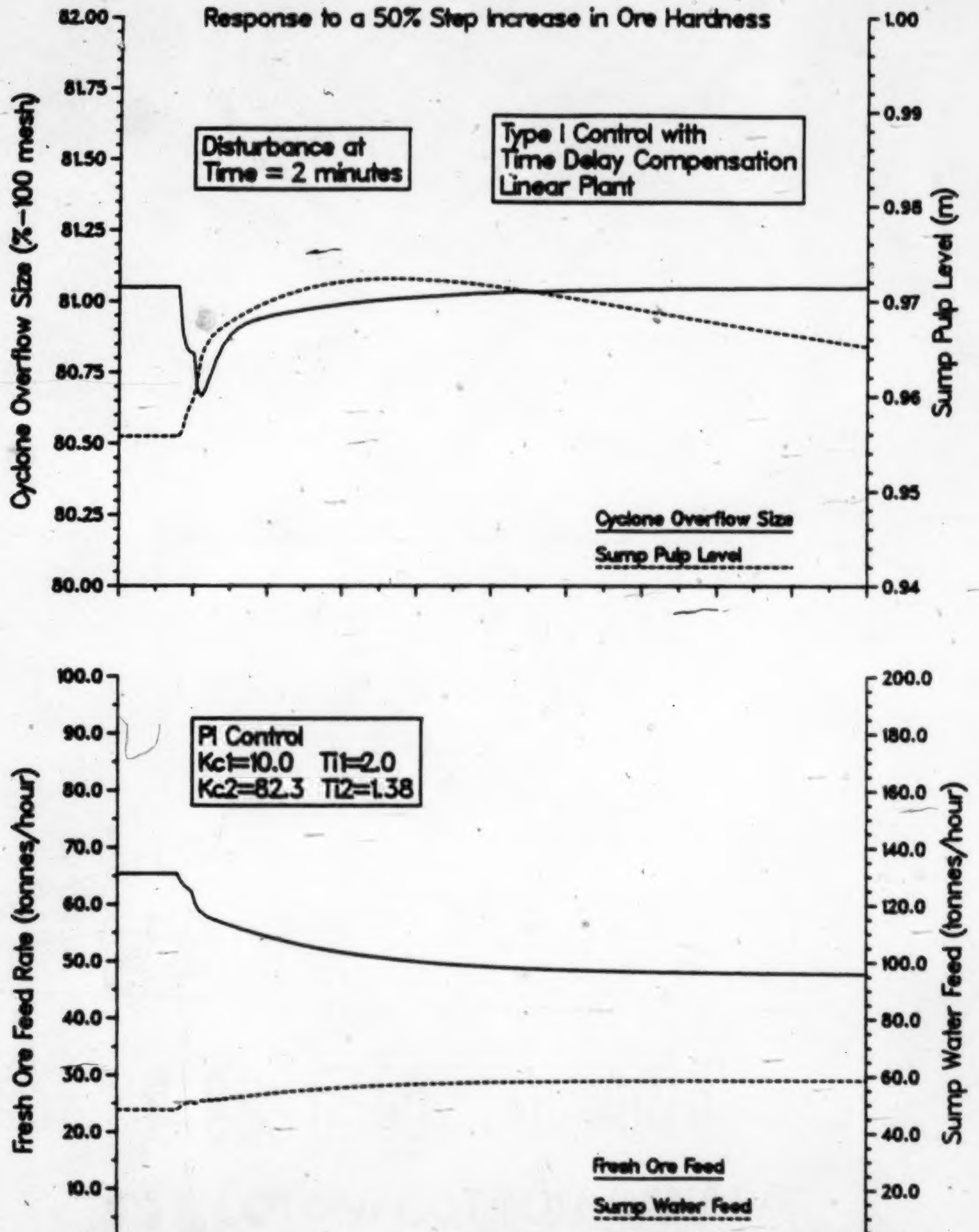


Figure 5.58: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation

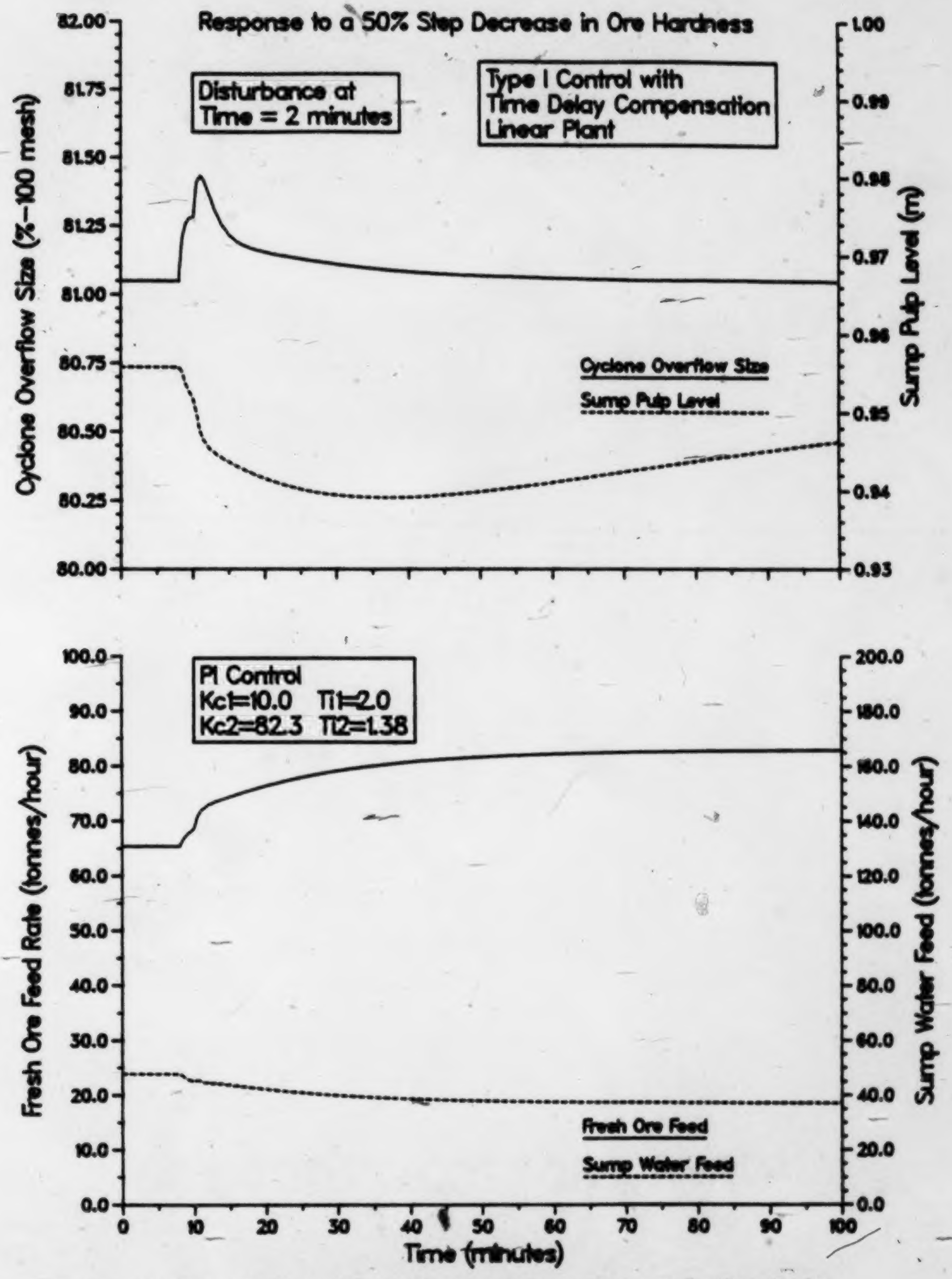


Figure 5.59: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Time Delay Compensation

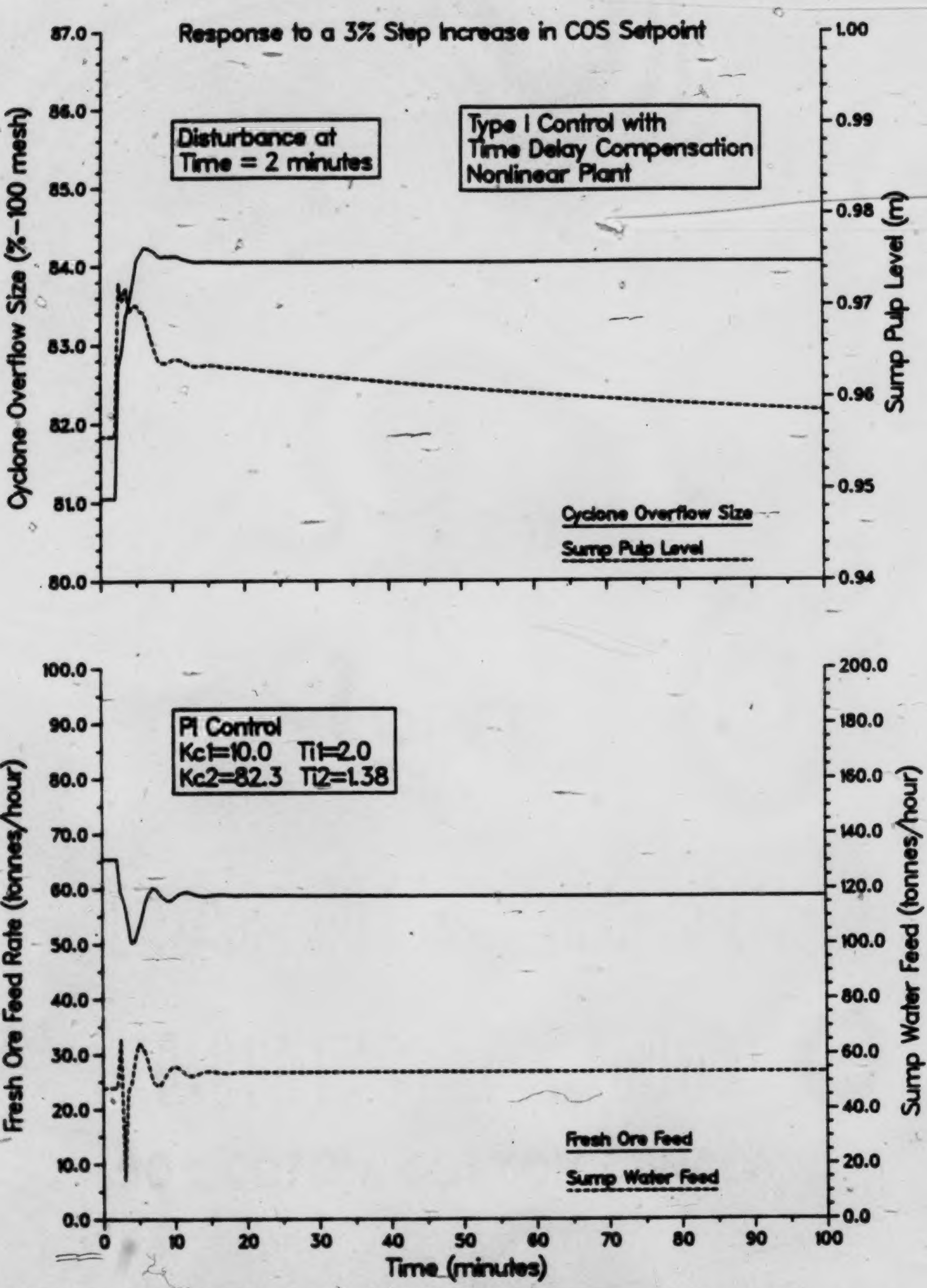


Figure 5.60: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation

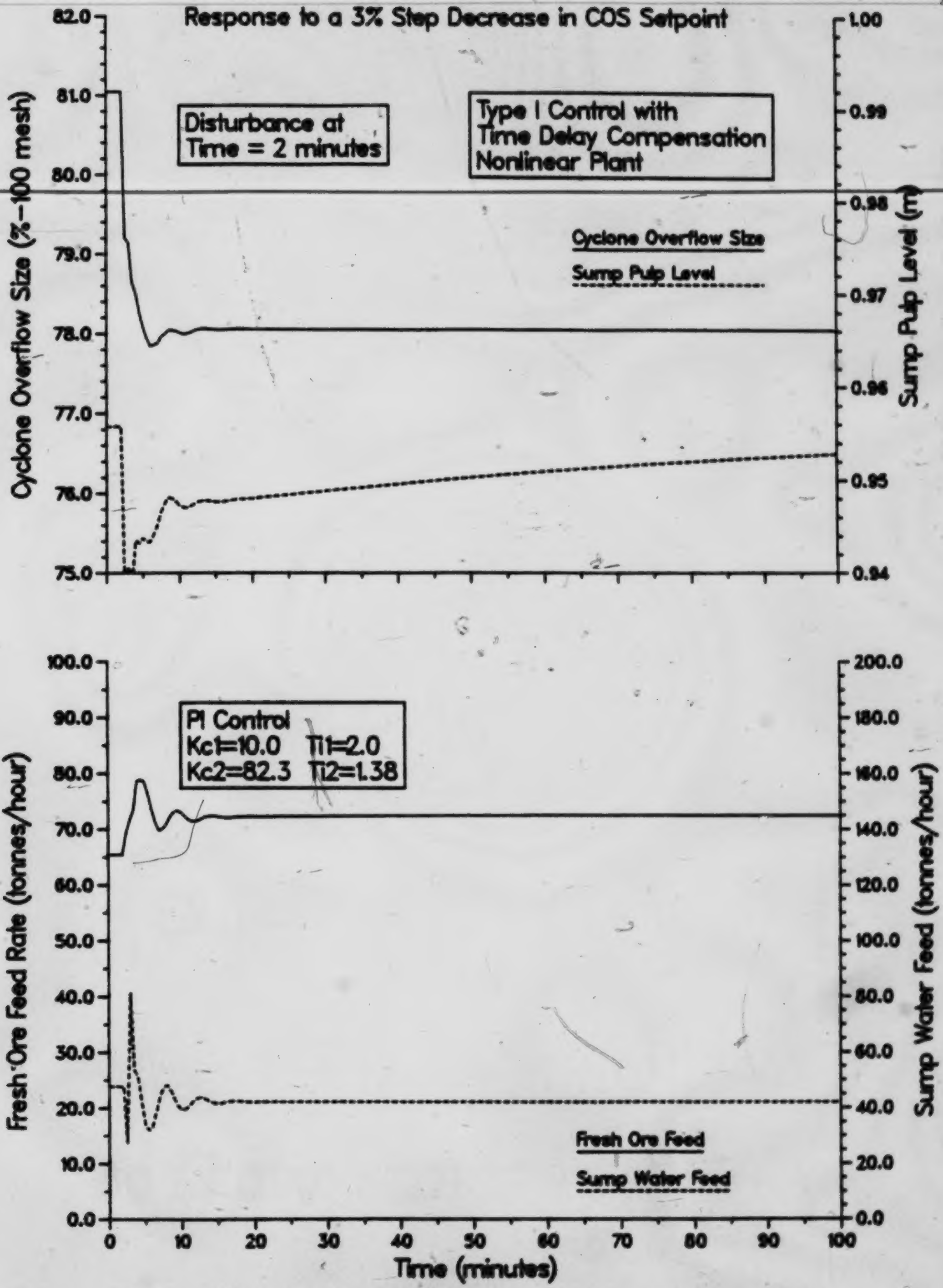


Figure 5.61: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation

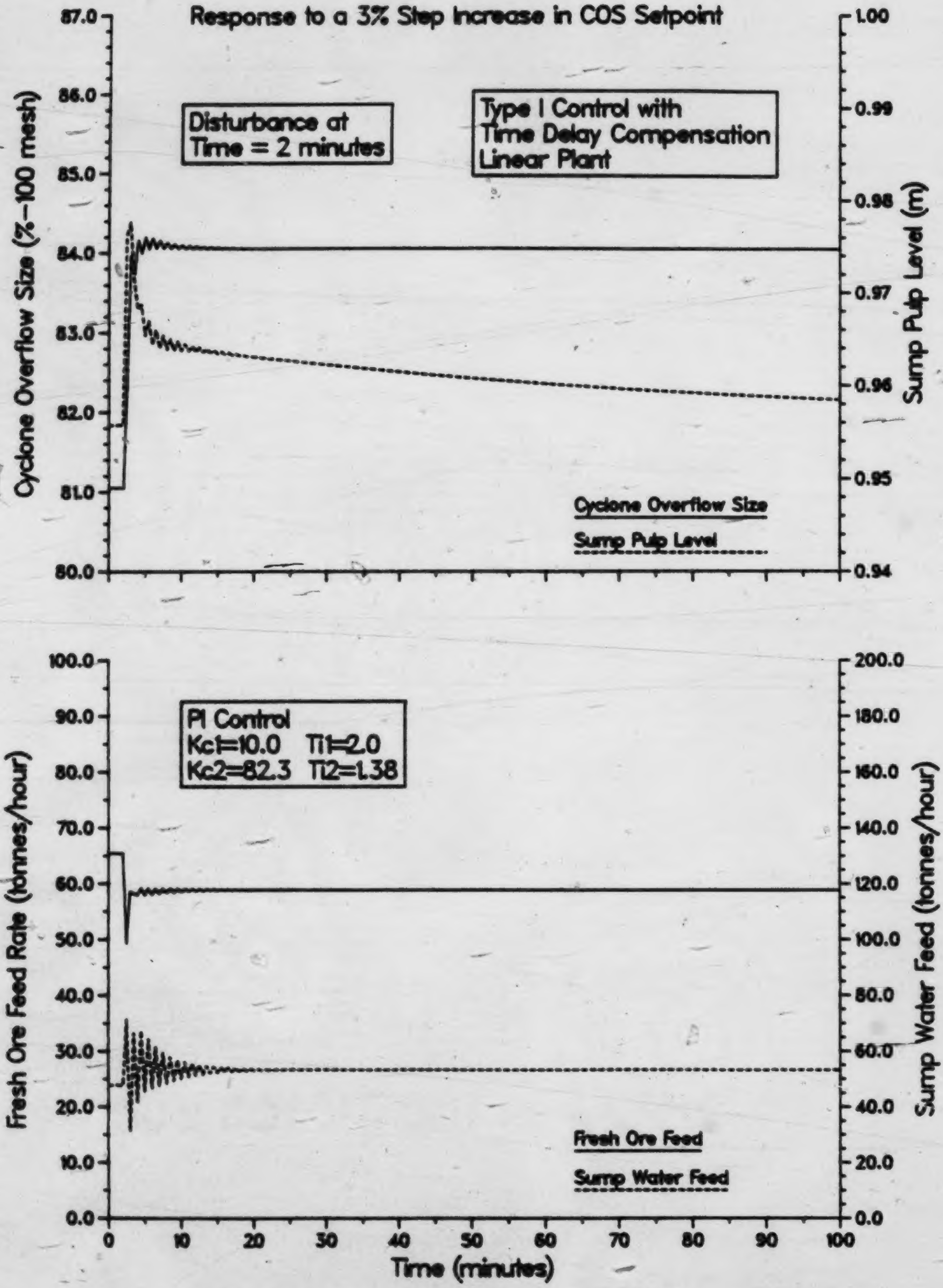


Figure 5.62: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation

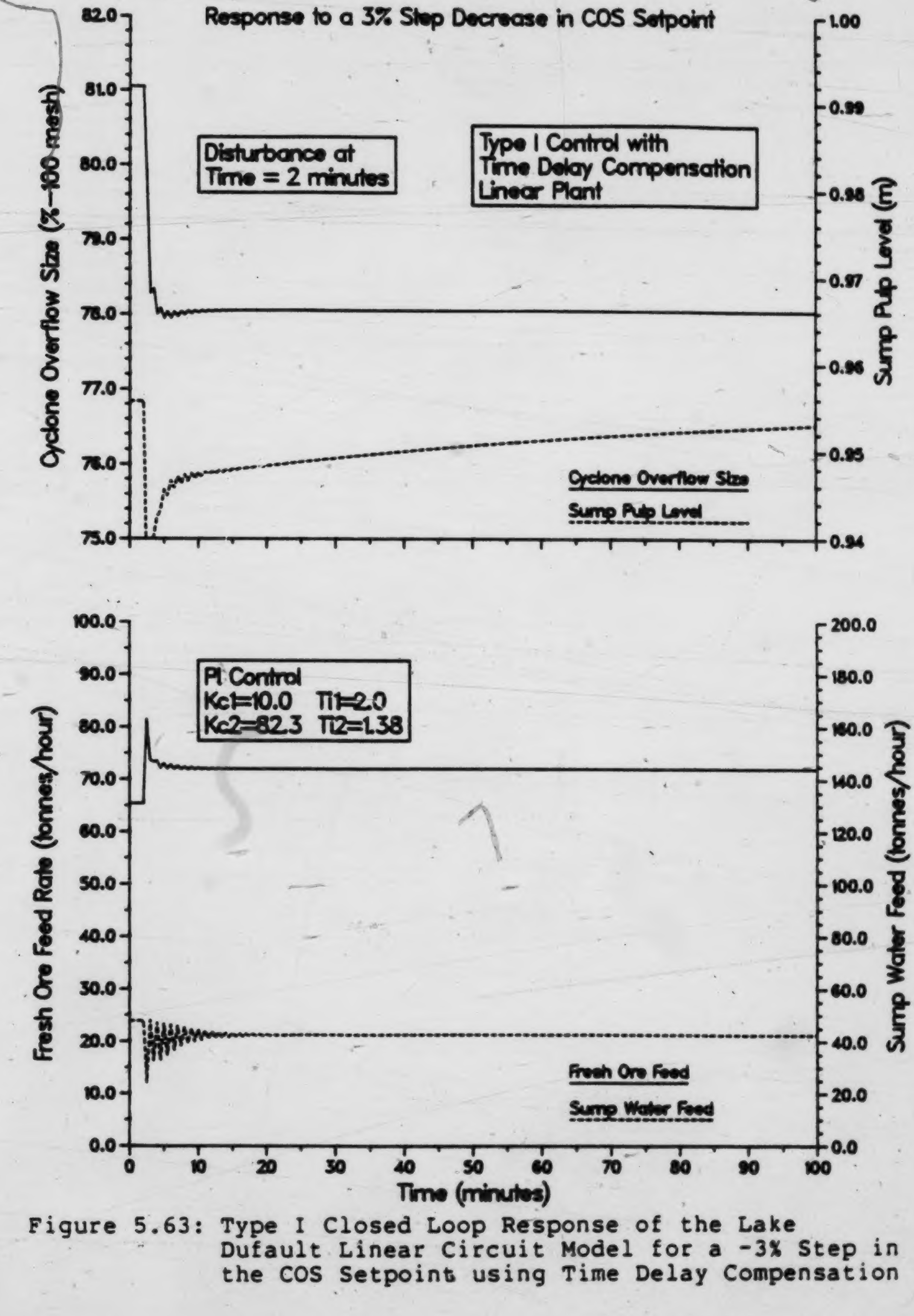


Figure 5.63: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation

those used in the uncompensated Type I simulation. By comparing Figures 5.60 and 5.61 to Figures 5.35 and 5.36, it can be seen that there has been a marked improvement in the setpoint tracking performance of loop 1 (COS) when using the Ogunnaike and Ray (1979) time delay compensation technique. The performance of loop 2, however, is degraded, albeit only slightly with the maximum deviation from the setpoint less than two centimetres which is quite acceptable for level control. It is interesting to note that the trends in the responses shown in Figures 5.62 and 5.63, using the linear model, although similar to those in Figures 5.60 and 5.61, obtained using the nonlinear model, show a slight improvement in the responses for loop 2 if the oscillations due to the time delay sensitivity of the linear model are ignored. This difference probably relates to a slight model mismatch between the linear and nonlinear grinding circuit models.

The benefits of time delay compensation are even more clearly demonstrated by the control performance shown in Figures 5.64 and 5.65 obtained using the linear model for the circuit with the controller gains adjusted to $K_{C1} = 15.0$ and $K_{C2} = 41.2$. For the 50% step increase in ore hardness, as can be seen from Figure 5.64, there has been only a marginal improvement in the disturbance rejection capabilities of the control system. On the other hand, the improvement in the setpoint tracking trajectory for loop 1 is clearly evident when Figure 5.65 is compared to Figure

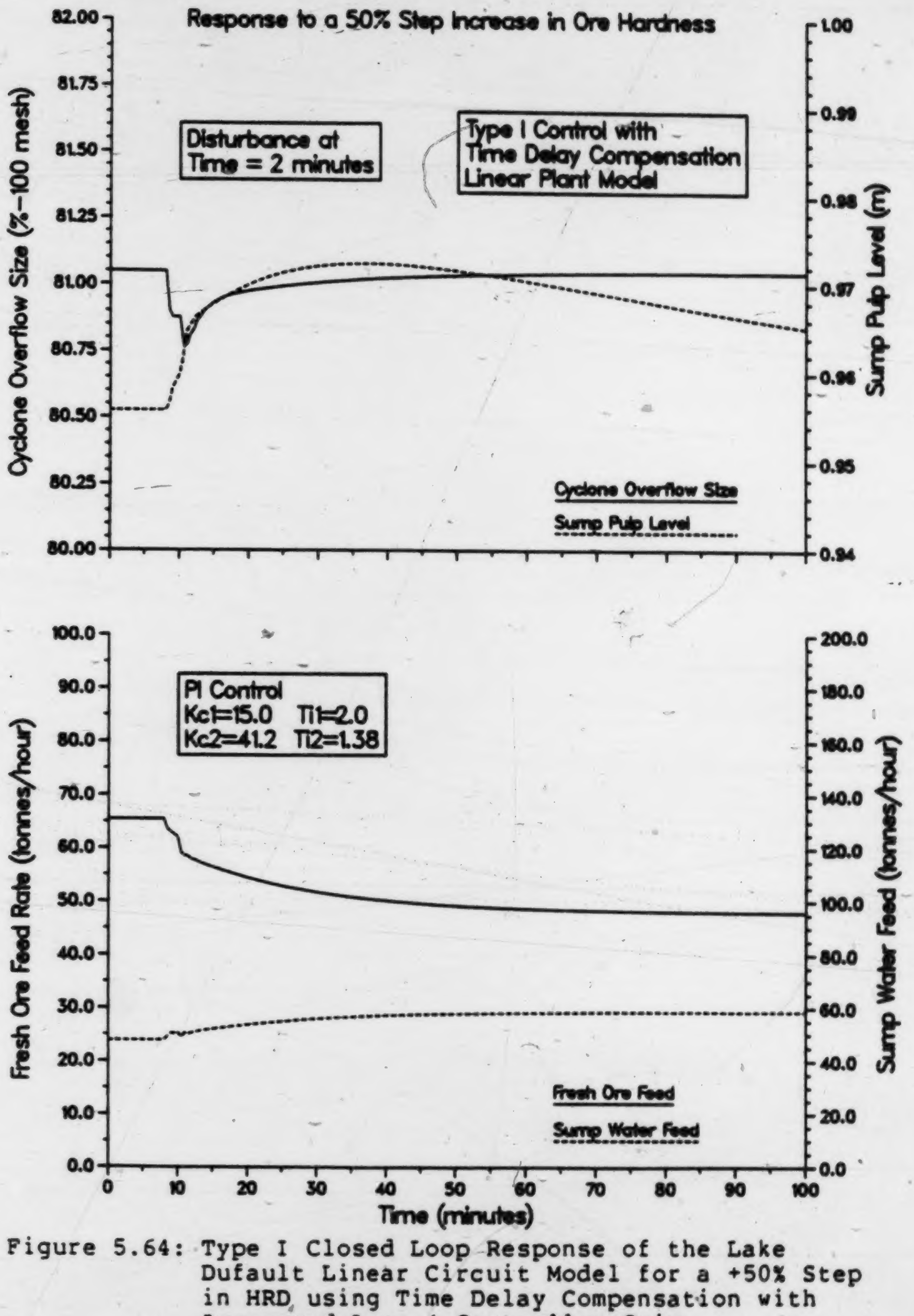


Figure 5.64: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation with Increased Loop 1 Controller Gain

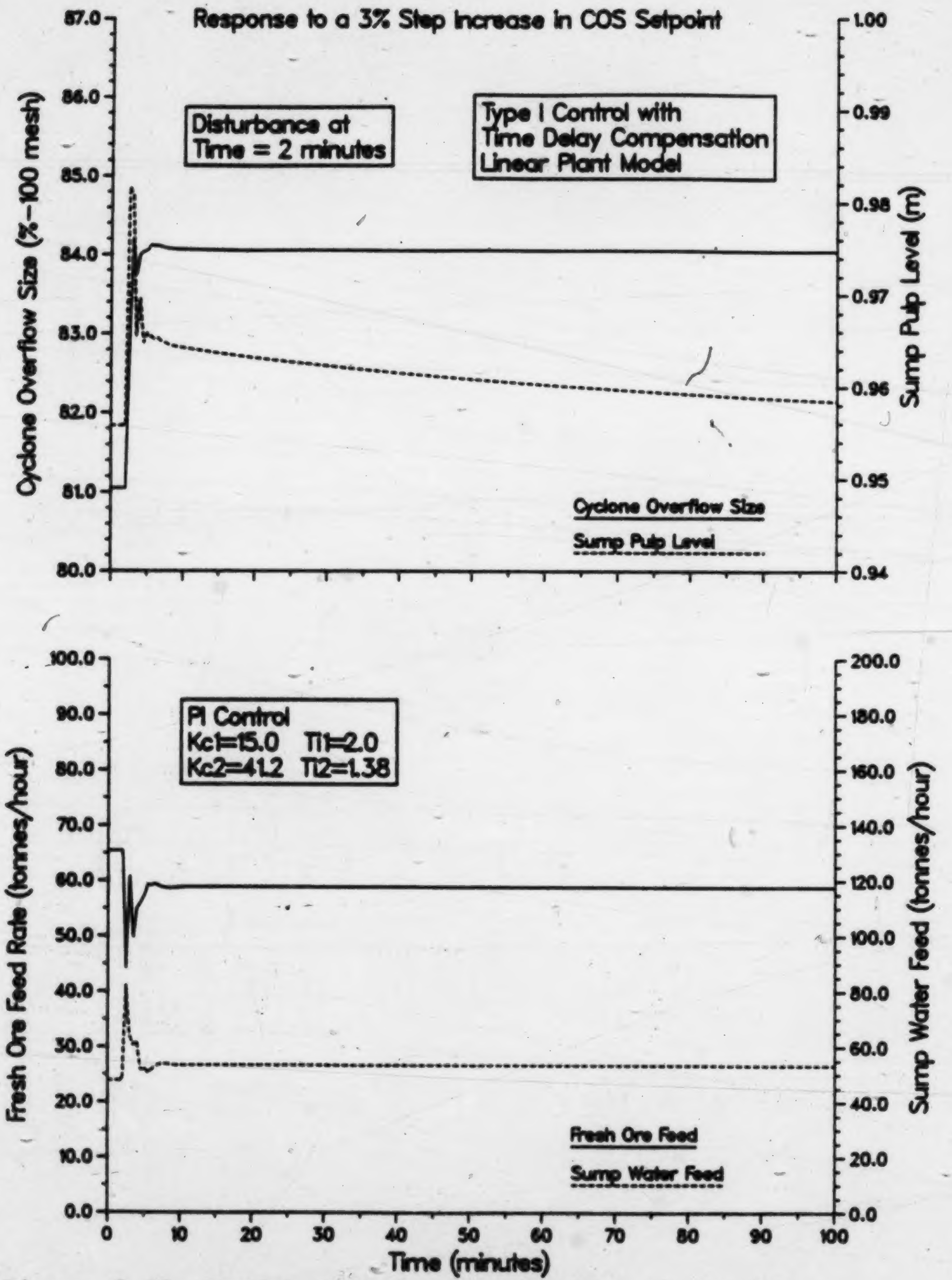


Figure 5.65: Type I Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation with Increased Loop 1 Controller Gain

5.62.

The simulations performed for the Type I variable pairing scheme using time delay compensation are summarized in Table 5.11 along with a brief comment on the control performance in each case.

The Type II configuration of the control system is even more sensitive to the choice of linear model time delays than the Type I scheme. For a 50% step increase in ore hardness as the disturbance input with the nonlinear model as the plant, it was necessary to reduce the loop 2 controller gain to $K_{C2} = 10.0$ in order to obtain the response shown in Figure 5.66. This is likely due to the additional interaction induced by the time delay compensator and the fact that the controller is in reality, controlling the linear model of the grinding circuit. Comparison of the responses in Figure 5.66 with the response given in Figure 5.45 indicates that there is some degradation of overall control system performance.

The results for a similar simulation using the linear model for the circuit are shown in Figure 5.67. The controller constants used in this simulation were the same as those used in Figure 5.66. Using the original Type II controller constants of $K_{C1} = 3.0$, $K_{C2} = 20.6$, and $\tau_I = 2.0$, $\tau_I = 5.0$ from the uncompensated linear model simulation produces the responses shown in Figure 5.68. Although these responses appear to be similar in nature to that given in Figure 5.66, an improvement in the control system

Table 5.11: Summary of the Simulations for the Lake Dufault Grinding Circuit using Time Delay Compensation and the Type I Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.56	-10.0	2.0	82.3	1.38	+50% step in hardness. Nonlinear circuit model. No improvement in loop 1 performance when compared with Fig. 5.37.
5.57	-10.0	2.0	82.3	1.38	-50% step in hardness. Nonlinear circuit model. No improvement in loop 1 performance when compared with Fig. 5.38.
5.58	-10.0	2.0	82.3	1.38	+50% step in hardness. Linear circuit model. Loop 1 performance indicates model mismatch.
5.59	-10.0	2.0	82.3	1.38	-50% step in hardness. Linear circuit model. Loop 1 performance indicates model mismatch.
5.60	-10.0	2.0	82.3	1.38	+3% step in COS setpoint. Nonlinear circuit model. Marked improvement in loop 1 performance when compared with Fig.5.35.

Table 5.11 continued.

5.61	-10.0	2.0	82.3	1.38	-3% step in COS setpoint. Nonlinear circuit model. Marked improvement in loop 1 performance when compared with Fig.5.36.
5.62	-10.0	2.0	82.3	1.38	+3% step in COS setpoint. Linear circuit model. Slight improvement in loop 2 performance when compared with Fig.5.60.
5.63	-10.0	2.0	82.3	1.38	-3% step in COS setpoint. Linear circuit model. Slight improvement in loop 2 performance when compared with Fig.5.61.
5.64	-15.0	2.0	41.2	1.38	+50% step in hardness. Linear circuit model. No improvement in control performance.
5.65	-15.0	2.0	41.2	1.38	+3% step in COS setpoint. Linear circuit model. Great improvement in loop 1 performance when compared with Fig.5.62.

performance can be seen if the response of the uncompensated system given in Figure 5.52 is considered.

The response of the nonlinear and linear models to a negative 50% step change in the ore hardness show similar trends, as illustrated by Figures 5.69 and 5.70. As with the positive hardness step, there is a degradation in overall control system performance compared with the

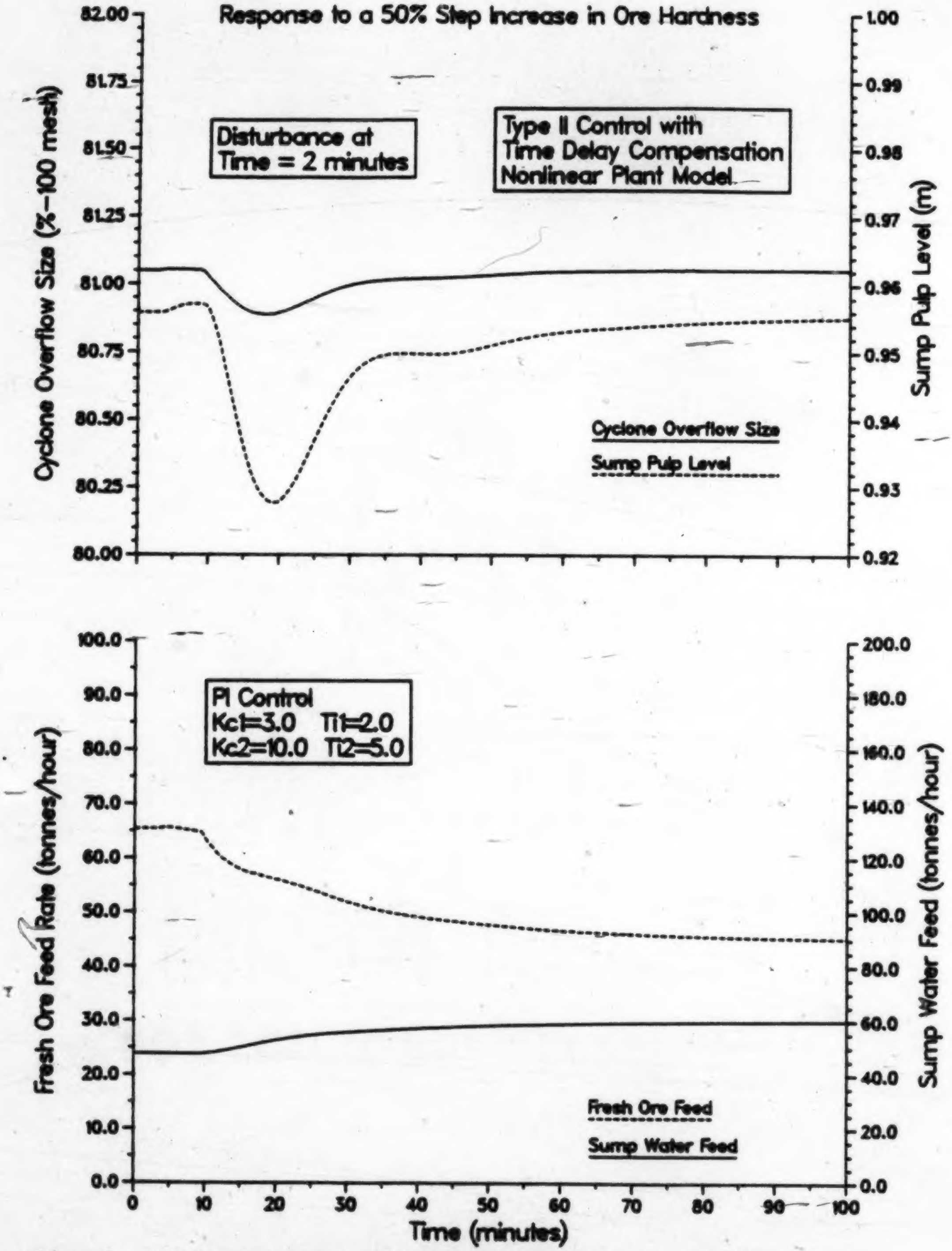


Figure 5.66: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Time Delay Compensation

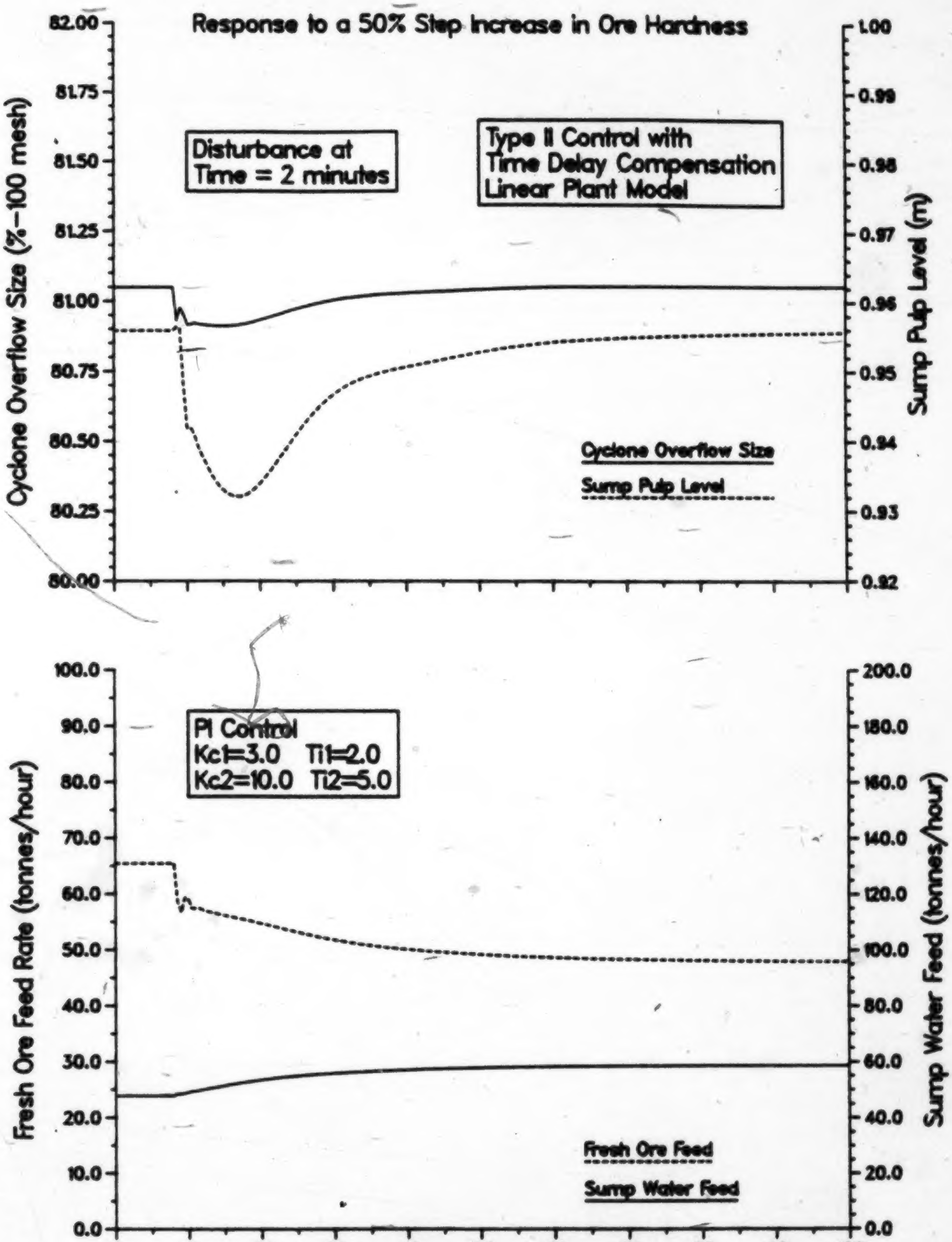


Figure 5.67: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation

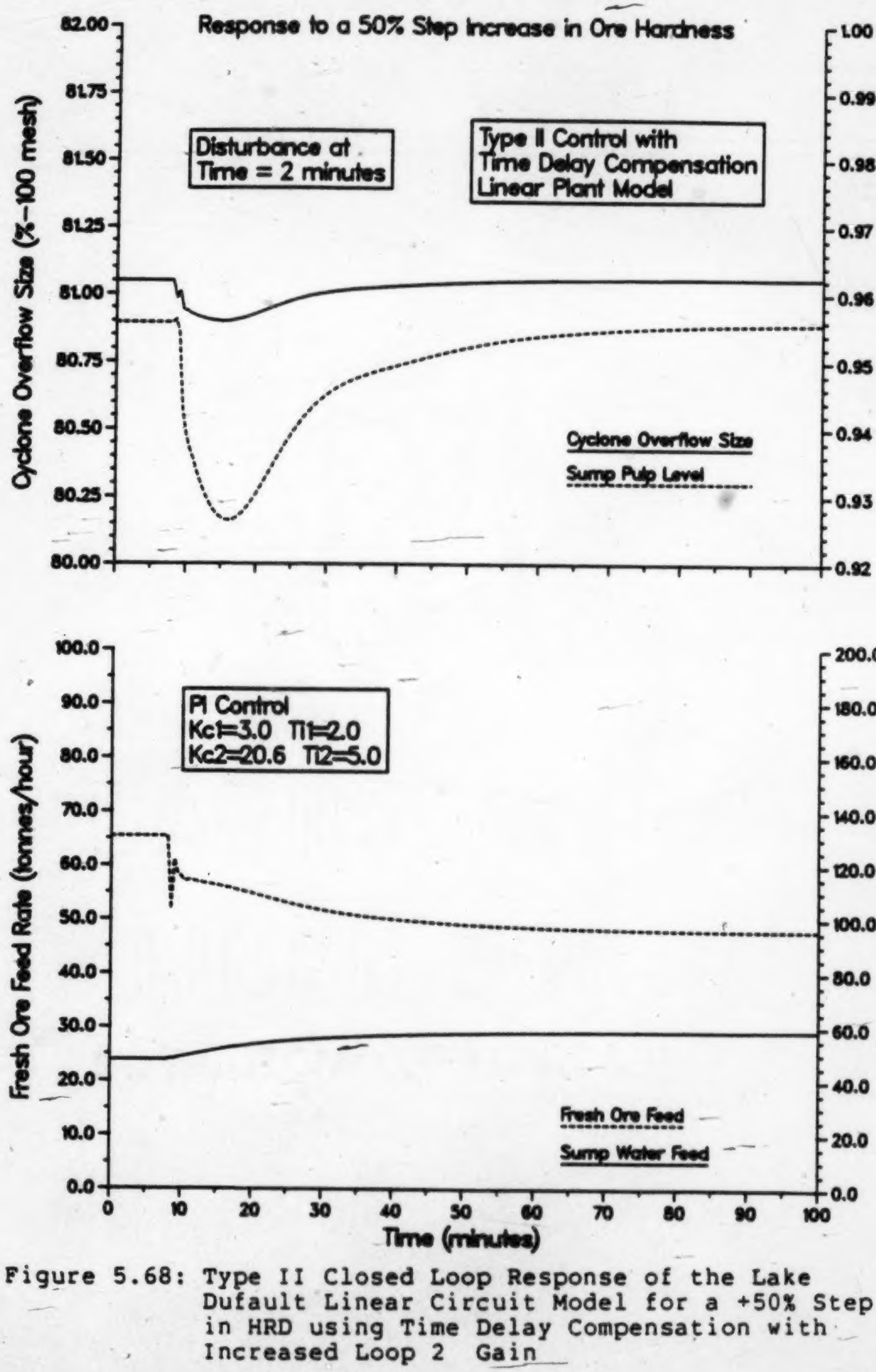


Figure 5.68: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +50% Step in HRD using Time Delay Compensation with Increased Loop 2 Gain

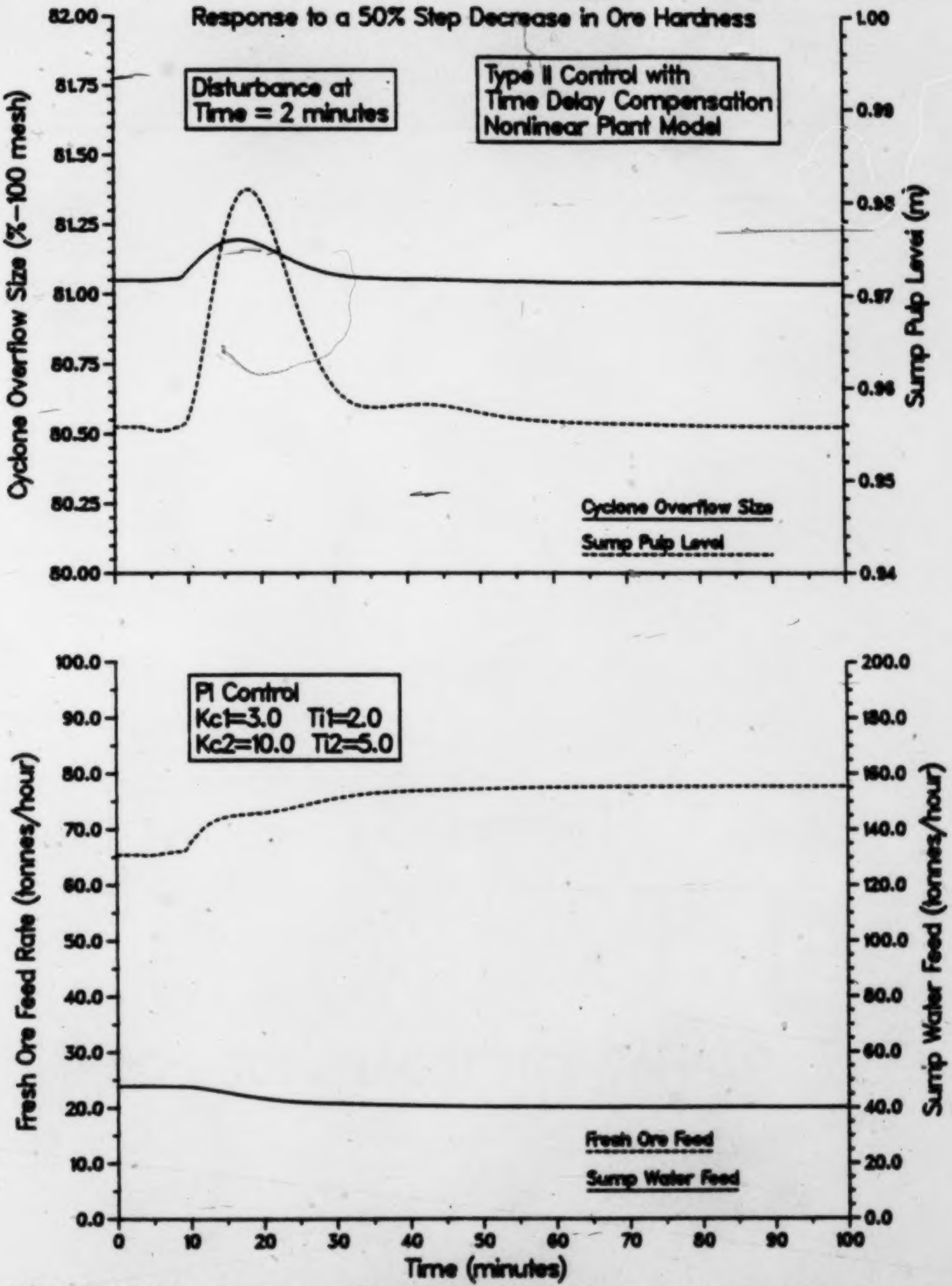


Figure 5.69: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -50% Step in HRD using Time Delay Compensation

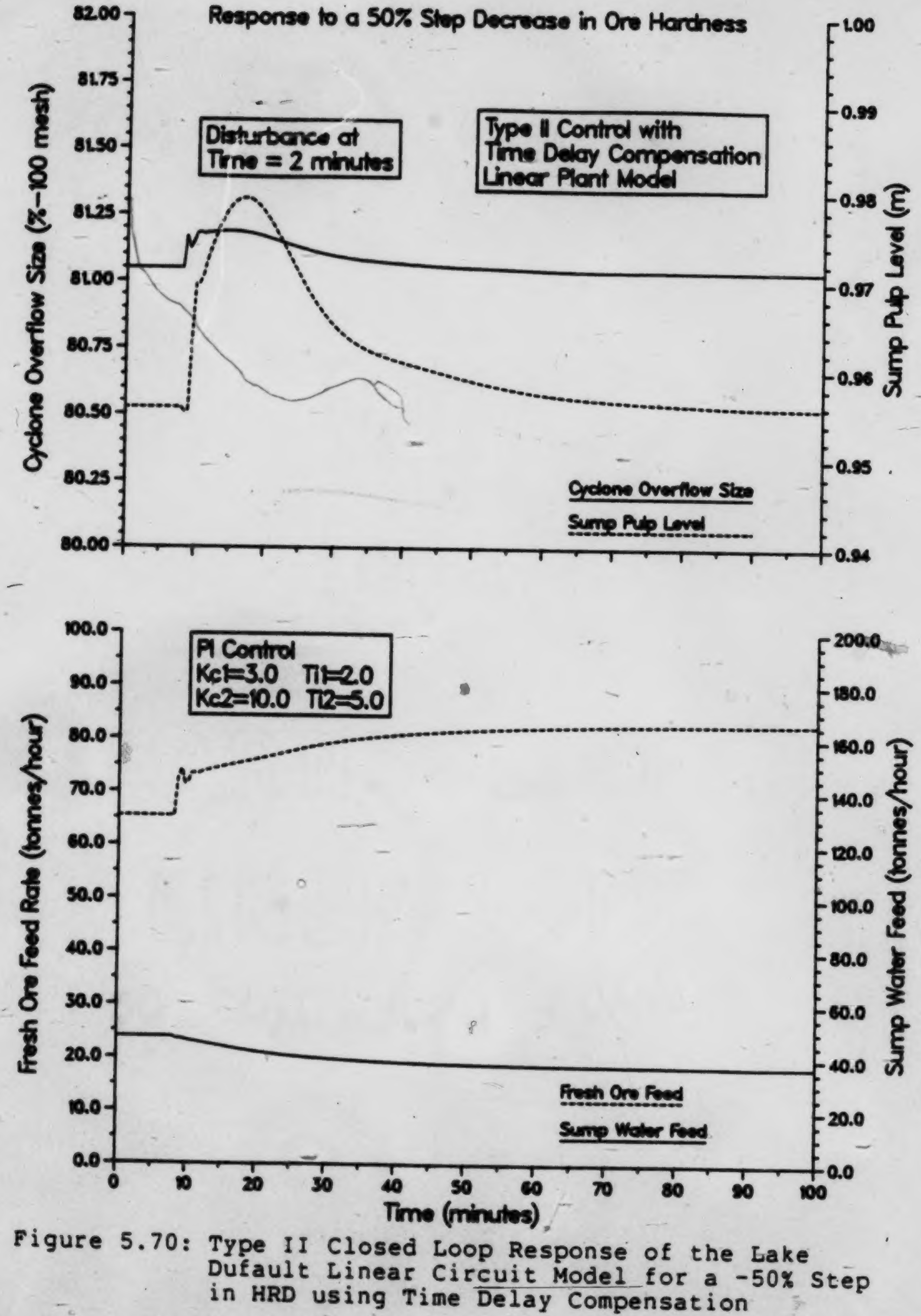


Figure 5.70: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -50% Step in HRD using Time Delay Compensation

uncompensated response for the nonlinear model (cf. Figure 5.46), while there is an improvement in performance when representing the circuit by the linear model (cf. Figure 5.53).

Simulation of the Type II control scheme using the nonlinear and linear models for positive and negative 3% step changes in the COS setpoint resulted in the responses presented in Figures 5.71 through 5.75. As was the case for the step changes in ore hardness, the loop 2 controller gain for the nonlinear plant simulation required detuning to obtain stable responses. Improvements in the performance of the control system over the uncompensated case is evident for both the nonlinear and linear plant models, as was found from the Type I results. The linear plant model demonstrates the most improvement as would be expected because of the absence of any mismatch between the plant model and the time delay compensator model.

The simulations performed for the Type II variable pairing scheme using time delay compensation are summarized in Table 5.12 along with a brief comment on the control performance in each case.

5.9 Multivariable Control

The results presented in Section 5.6 illustrate the fact that multiloop control, in the presence of interaction, leaves much to be desired. It should be clear that if the interaction in either of the Type I or Type II control

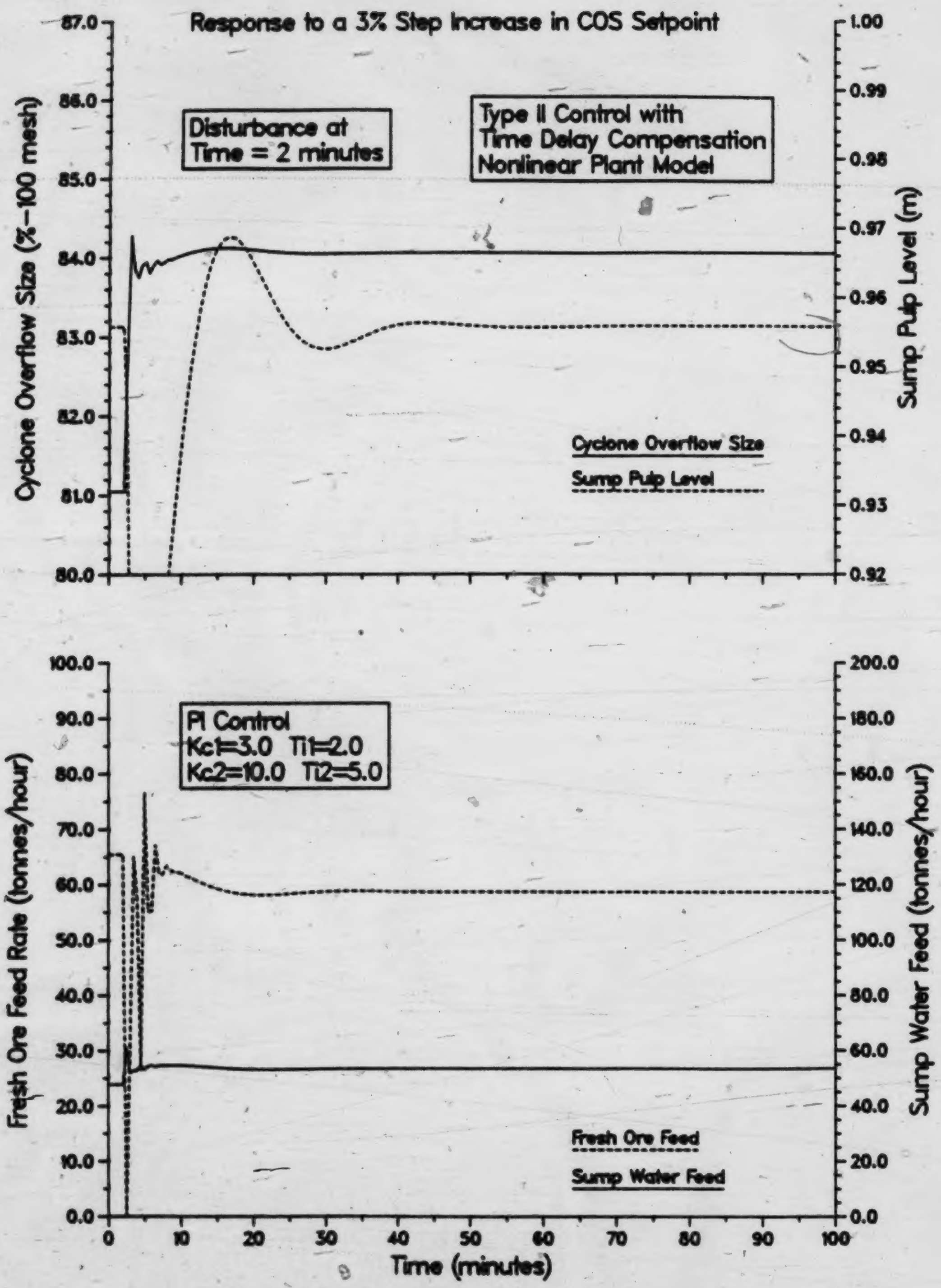


Figure 5.71: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation

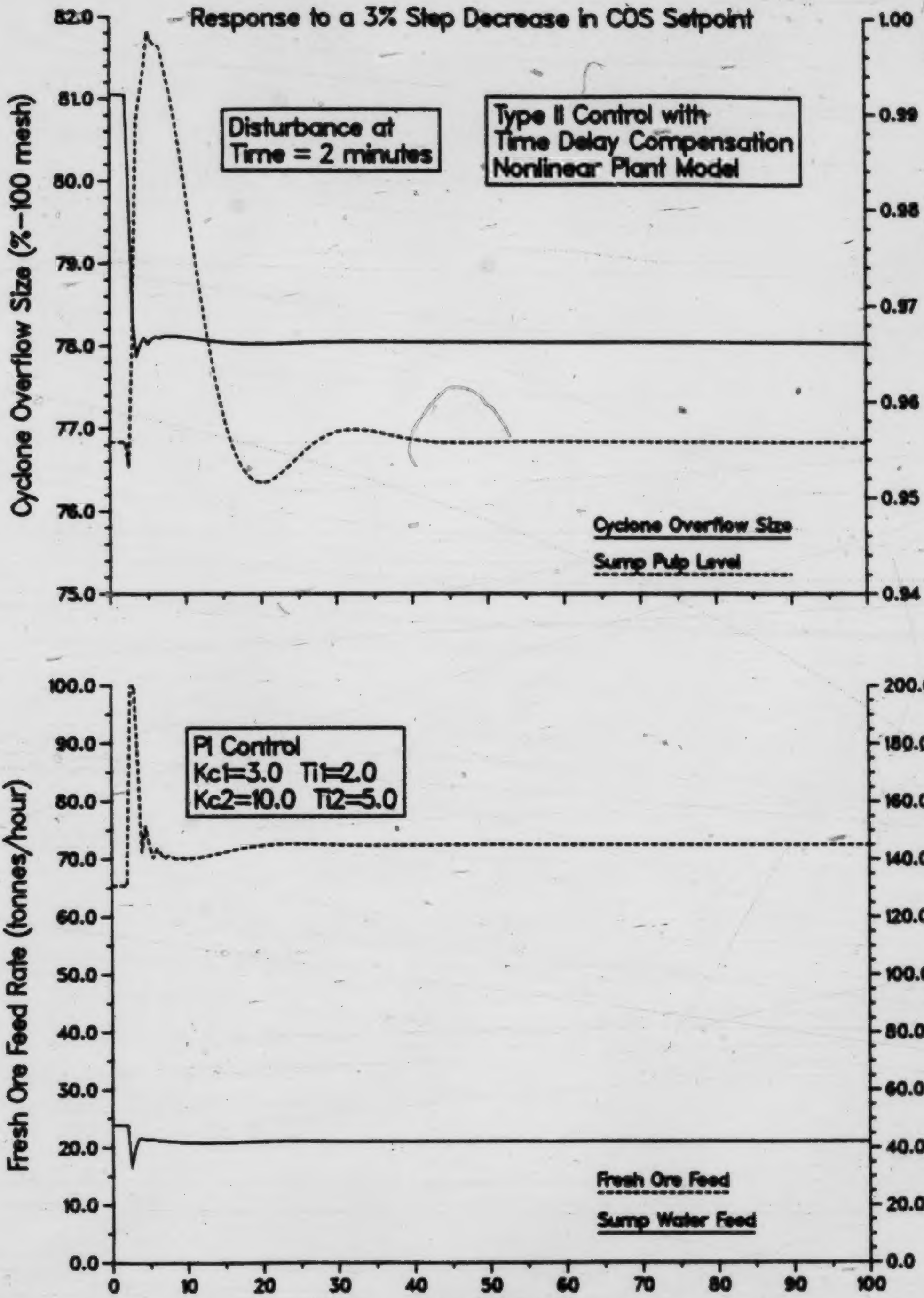


Figure 5.72: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation

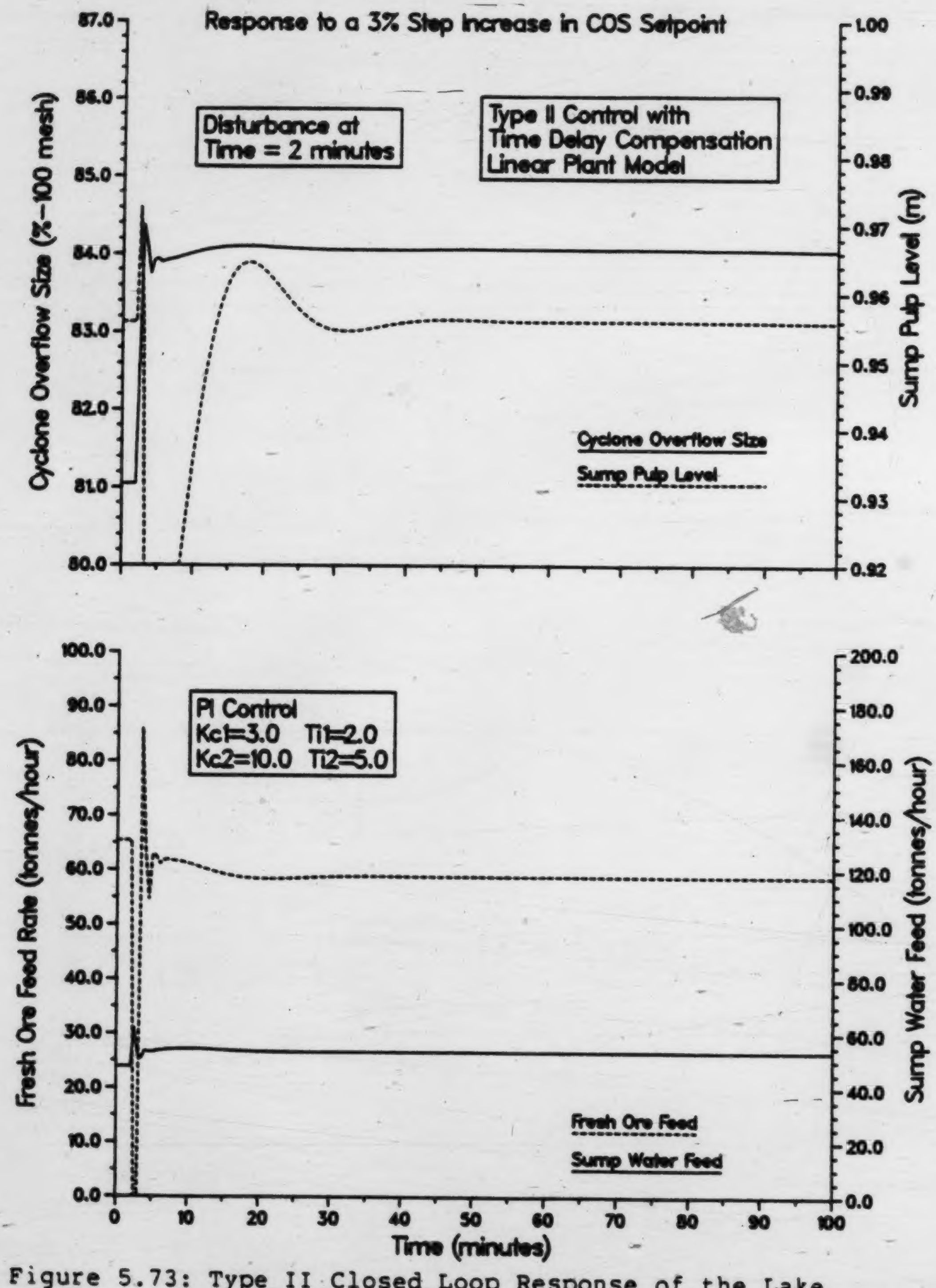


Figure 5.73: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation

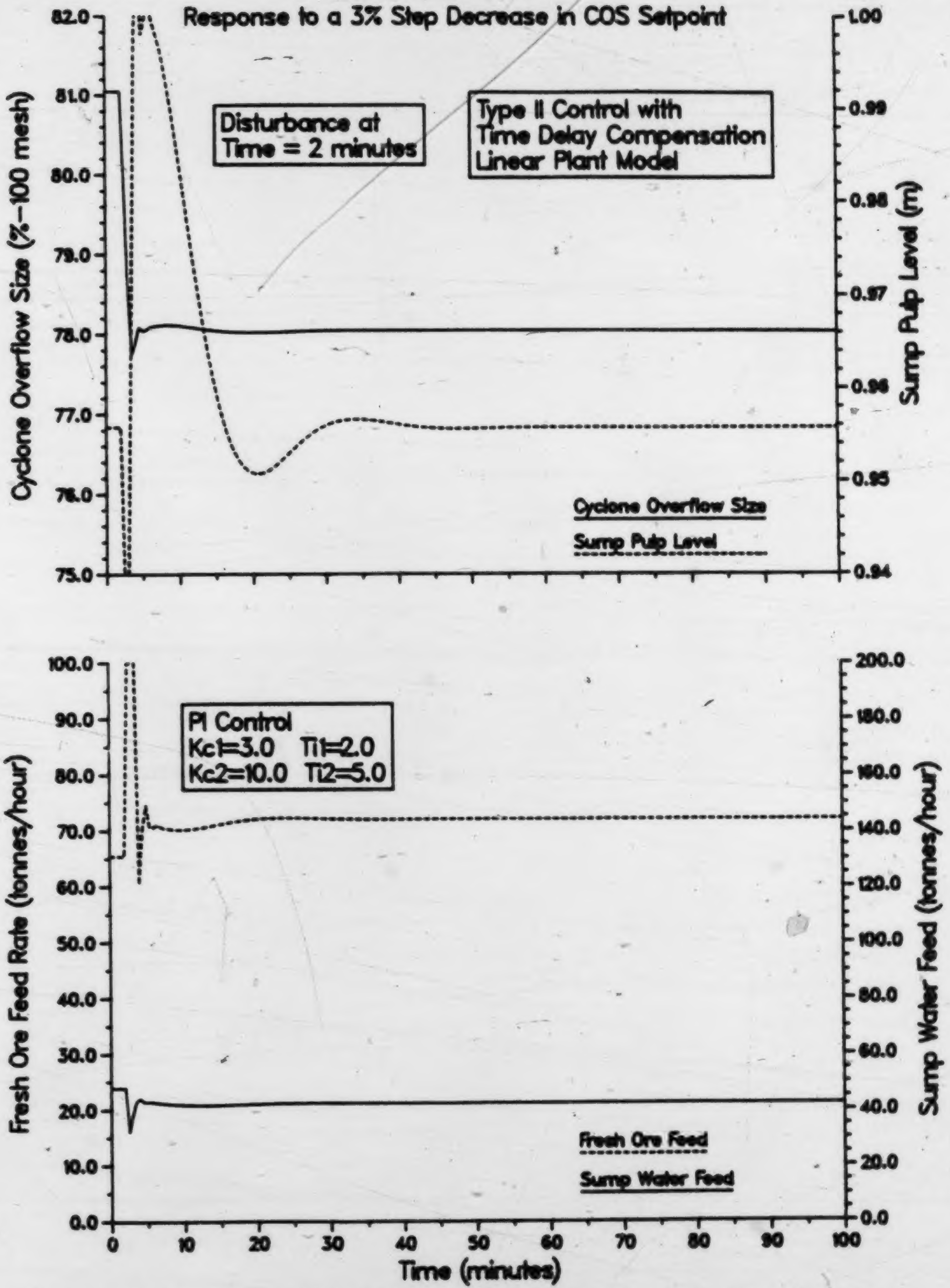


Figure 5.74: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a -3% Step in the COS Setpoint using Time Delay Compensation

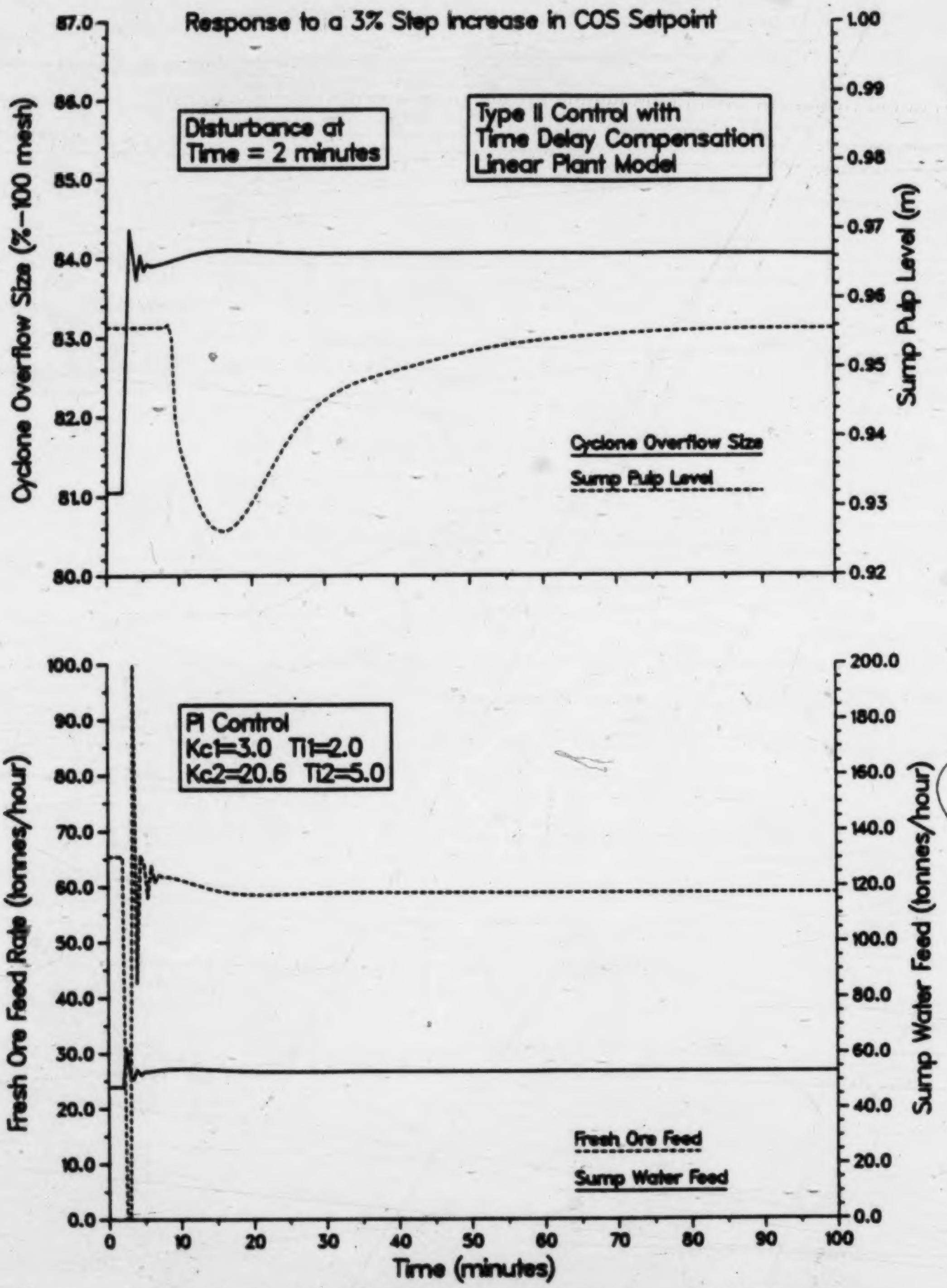


Figure 5.75: Type II Closed Loop Response of the Lake Dufault Linear Circuit Model for a +3% Step in the COS Setpoint using Time Delay Compensation with Increased Loop 2 Controller Gain

Table 5.12: Summary of the Simulations for the Lake Dufault Grinding Circuit using Time Delay Compensation and the Type II Variable Pairing Scheme

Figure	Controller Constants				Comment
	K_{C1}	τ_{I1}	K_{C2}	τ_{I2}	
5.66	3.0	2.0	10.0	5.0	+50% step in hardness. Nonlinear circuit model. No improvement control performance when compared with Fig. 5.45.
5.67	3.0	2.0	10.0	5.0	+50% step in hardness. Linear circuit model. Slight improvement in control performance when compared with Fig. 5.66.
5.68	3.0	2.0	20.6	5.0	+50% step in hardness. Linear circuit model. Improvement in control performance compared with Fig. 5.52.
5.69	3.0	2.0	10.0	5.0	-50% step in hardness. Nonlinear circuit model. No improvement control performance when compared with Fig. 5.46.
5.70	3.0	2.0	10.0	5.0	-50% step in hardness. Linear circuit model. Slight improvement in performance when compared with Fig.5.69.

Table 5.12 continued.

5.71	3.0	2.0	10.0	5.0	+3% step in COS setpoint. Nonlinear circuit model. Improvement in loop 1 performance when compared with Fig.5.48.
5.72	3.0	2.0	10.0	5.0	-3% step in COS setpoint. Nonlinear circuit model. Improvement in loop 1 performance when compared with Fig.5.49.
5.73	3.0	2.0	10.0	5.0	+3% step in COS setpoint. Linear circuit model. Slight improvement in control performance when compared with Fig.5.71.
5.74	3.0	2.0	10.0	5.0	-3% step in COS setpoint. Linear circuit model. Slight improvement in control performance when compared with Fig.5.72.
5.75	3.0	2.0	20.6	5.0	+3% step in COS setpoint. Linear circuit model. Slight improvement in control performance when compared with Fig.5.73.

schemes can be eliminated, then better overall system performance could be achieved.

5.9.1 Interaction Analysis

The first step toward possible improvement in control system performance is the use of interaction analysis, as described in Chapter 2 (cf. Section 2.6.2) to examine the choice of the pairing of manipulated and controlled variables. The Type I variable pairing is postulated initially so that the steady state process gain matrix (cf. equation (2.59)) is:

$$\mathbf{A} = \begin{bmatrix} -0.179 & 0.337 \\ 0.161 & 0.199 \end{bmatrix} \tag{5.25}$$

It should be noted that this matrix is derived directly from the linear model discussed in Section 5.4. Inverting the matrix in equation (5.25) and taking the transpose yields:

$$\mathbf{B} = \begin{bmatrix} -2.214 & 1.791 \\ 3.750 & 1.992 \end{bmatrix} \tag{5.26}$$

The individual elements of the relative gain array (RGA) are calculated from equations (5.25) and (5.26) using equation (2.55), which results in:

$$\text{RGA} = \begin{array}{c|cc} & \text{FOF} & \text{SWF} \\ \hline \text{COS} & 0.396 & 0.604 \\ \text{SPL} & 0.604 & 0.396 \end{array} \tag{5.27}$$

It can be seen from the RGA that the postulated pairing (Type I) does not give the least amount of steady state interaction. It is to be noted from the RGA that the interaction with the Type II pairing is still quite strong. As well, the Type II pairing will put the slowest dynamics and the longest time delays on the direct transmission paths in both control loops. This may not be desirable, and in fact, will lead to complications when designing decoupling controllers with time delay elements.

5.9.2 Decoupler Design

Although the interaction analysis presented in the previous section indicated that the optimal pairing resulted in a Type II control scheme, it is instructive to examine the control performance of both choices of variable pairing using a decoupling control strategy since both control schemes are used in industry. Using the transfer function model representation of the Lake Dufault grinding circuit (see section 5.4), it is easy to design decoupling controllers in the same manner as for the idealized system discussed in Appendix E. Both static decoupling and Zalkind's dynamic decoupler will be discussed here because, as mentioned in Chapter 2 (cf. Section 2.6.3), Luyben (1970) found no advantage to using an ideal decoupler. Furthermore, the FORTRAN code for implementation of Boksenbom and Hood's (1949) decoupler is very similar to that used for the Zalkind approach.

The static decoupling controllers are calculated from the steady state process gains of the transfer function model of the grinding circuit using equations (2.80) and (2.81). For the Type I scheme, the static decoupling controller matrix is:

$$\mathbf{D} = \begin{bmatrix} 1.0 & 1.88 \\ -0.809 & 1.0 \end{bmatrix} \tag{5.28}$$

and for the Type II scheme, the decoupler is:

$$\mathbf{D} = \begin{bmatrix} 1.0 & 0.531 \\ -1.24 & 1.0 \end{bmatrix} \tag{5.29}$$

Block diagrams for these noninteracting control schemes are given in Figures 5.76 and 5.77.

Zalkind's dynamic decoupler for the Type I control scheme is shown in the block diagram in Figure 5.78. From the linear model (cf. Table 5.5, Section 5.4) and equation (2.80) and (2.81), the controller to decouple loop 1 from loop 2 is:

$$G^*_{C12} = \frac{1.88(4.185s+1)e^{-.18s}}{2.32s+1} \tag{5.30}$$

and that to decouple loop 2 from loop 1 is:

$$G^*_{C21} = \frac{-0.809(80.0s+1)e^{-0.52s}}{77.6s+1} \tag{5.31}$$

It can be seen that the resultant controllers are first order lead/lag compensators combined with a time delay

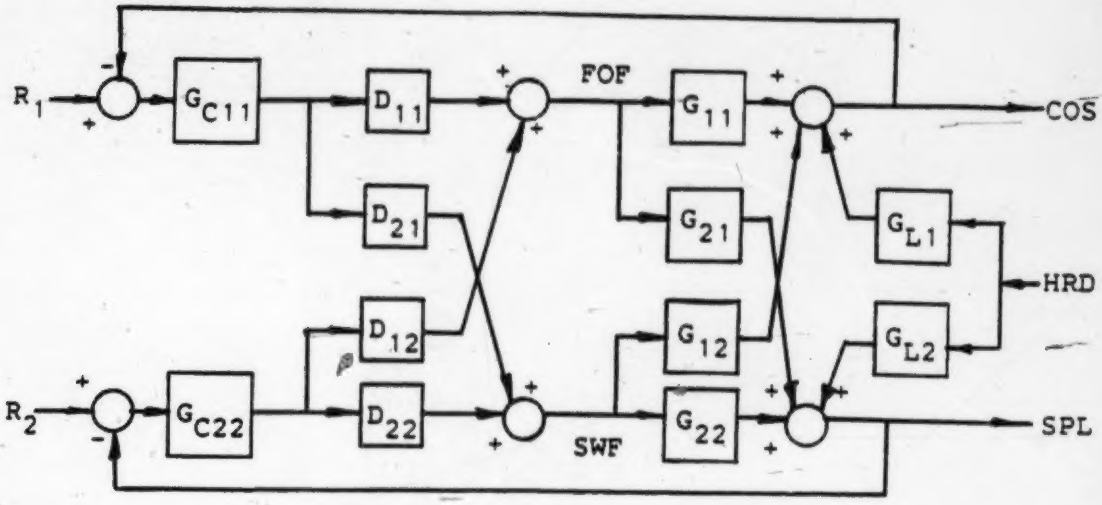

R1
R2
GC11
GC22
D11
D21
D12
D22
FOF
SWF
G11
G21
G12
G22
GL1
GL2
HRD
COS
SPL
+
-

Figure 5.76: Block Diagram Showing the Type I Static Decoupling Control System

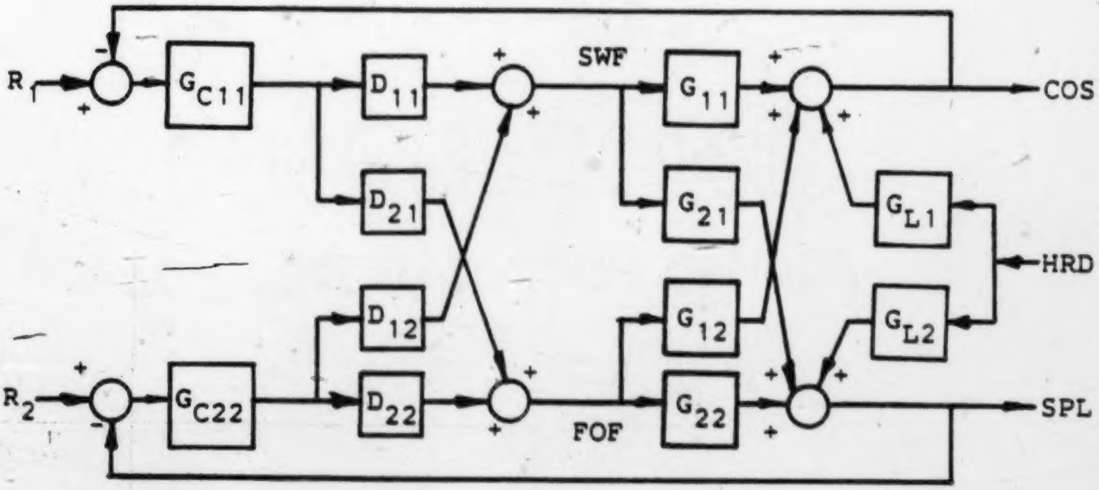

R1
R2
GC11
GC22
D11
D21
D12
D22
SWF
FOF
G11
G21
G12
G22
GL1
GL2
HRD
COS
SPL
+
-

Figure 5.77: Block Diagram Showing the Type II Static Decoupling Control System

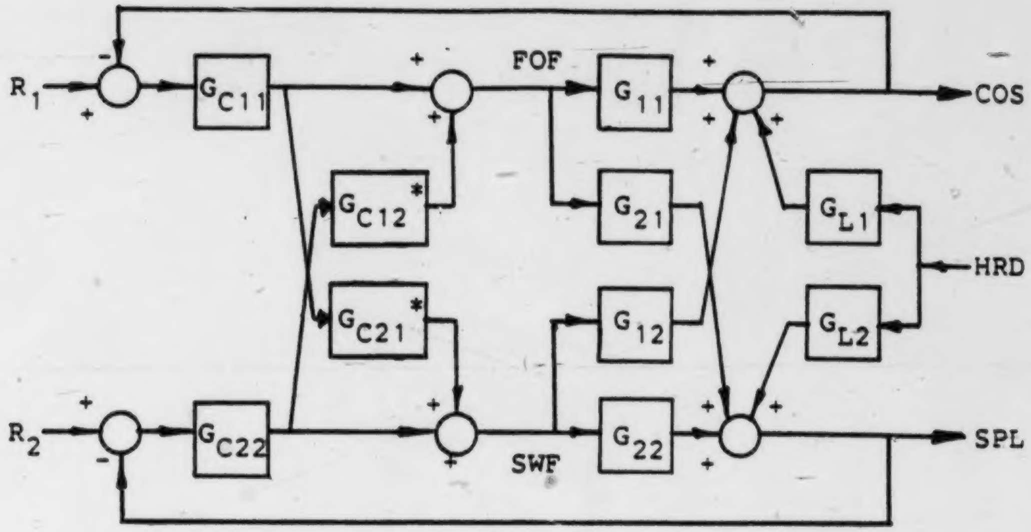


Figure 5.78: Block Diagram Showing the Type I Noninteracting Control System Configuration

term. A complete summary of the transfer functions used to establish these controllers is given in Table 5.13. The Type II decoupling controllers are also given in this table. The block diagram for dynamic decoupling of the Type II system is given in Figure 5.79.

It is obvious there is a problem with both decoupling controllers, G^*_{C12} and G^*_{C21}, for the Type II configuration since the positive exponential terms are unrealizable because this would require predictions into the future. In the evaluation of decoupling action for the Type II system these terms of the decoupling controllers are ignored.

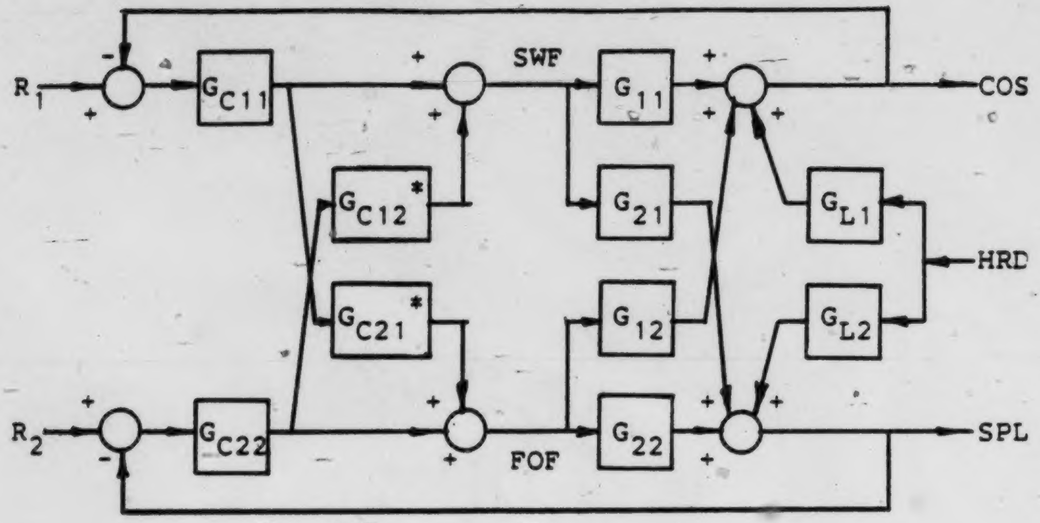


Figure 5.79: Block Diagram Showing the Type II Noninteracting Control System Configuration

5.10 Noninteracting Control Behavior

The interacting Type I and II control systems investigated in Section 5.6 will now be decoupled through the use of the noninteracting control strategies discussed in Section 5.9.

Before attempting to apply these schemes to the grinding circuit, various aspects of decoupling two by two systems were examined using an arbitrary, simple linear model. A discussion of this analysis can be found in Appendix E. The actual grinding circuit model is not as easy to decouple as the model used in the appendix because the nonlinear model, as the name implies, does not have a linear response to all

Table 5.13: First Order Transfer Function Model Representation of the Lake Dufault Grinding Circuit Including the Decoupling Controllers

	Type I	Type II
G_{11}	$\frac{-0.179}{4.18s+1}$	$\frac{0.337e^{-0.18s}}{2.32s+1}$
G_{12}	$\frac{0.337e^{-0.18s}}{2.32s+1}$	$\frac{-0.179}{4.18s+1}$
G_{21}	$\frac{0.161e^{-0.62s}}{77.6s+1}$	$\frac{0.199e^{-0.1s}}{80.1s+1}$
G_{22}	$\frac{0.199e^{-0.1s}}{80.1s+1}$	$\frac{0.161e^{-0.62s}}{77.6s+1}$
L_1	$\frac{-0.136e^{-6.0s}}{17.6s+1}$	$\frac{-0.136e^{-6.0s}}{17.6s+1}$
L_2	$\frac{0.0141e^{-8.0s}}{82.3s+1}$	$\frac{0.0141e^{-8.0s}}{82.3s+1}$
G^*_{C12}	$\frac{1.88(4.18s+1)e^{-.18s}}{2.32s+1}$	[1] $\frac{0.531(2.32s+1)e^{+.18s}}{4.185s+1}$
G^*_{C21}	$\frac{-0.809(80.1s+1)e^{-0.52s}}{77.6s+1}$	[1] $\frac{-1.24(77.6s+1)e^{+0.52s}}{80.1s+1}$

1. Delay will be ignored as predictive in nature.

inputs and poses special problems.

The manipulated and controlled variable responses of the nonlinear model using a Type I control scheme for a 50% step increase in the ore hardness are shown in Figure 5.80. This behavior resulted using controller constants of $K_{C1}=-10.0$, $\tau_{I1}=2.0$ minutes, $K_{C2}=82.3$, and $\tau_{I2}=1.38$ minutes the appropriate static decoupling controllers given by equations (5.28) and (5.29). It can be seen that this control performance using the same values as used for the Type I servo control simulations in Section 5.6 shows a small improvement over the responses shown in Figure 5.37. With dynamic decoupling it is expected that there should be an improvement in performance, however, using the decoupling controllers given by equations (5.30) and (5.31) an unstable response resulted for the same load disturbance. This behavior was attributed to the lead time dominance of both lead-lag units used as the decoupling controllers. By modifying the constants in the decoupling controllers, it is possible to stabilize the control system. Reducing the lead time constant in both decouplers to 1.0 minutes, and again simulating the control system response to a 50% step increase in ore hardness results in the response curves shown in Figure 5.81. A comparison of the responses with those obtained without decoupling controllers shows that the performance of the system is somewhat degraded. Attempts at further tuning of the constants in the decoupling controllers were unsuccessful in improving the performance

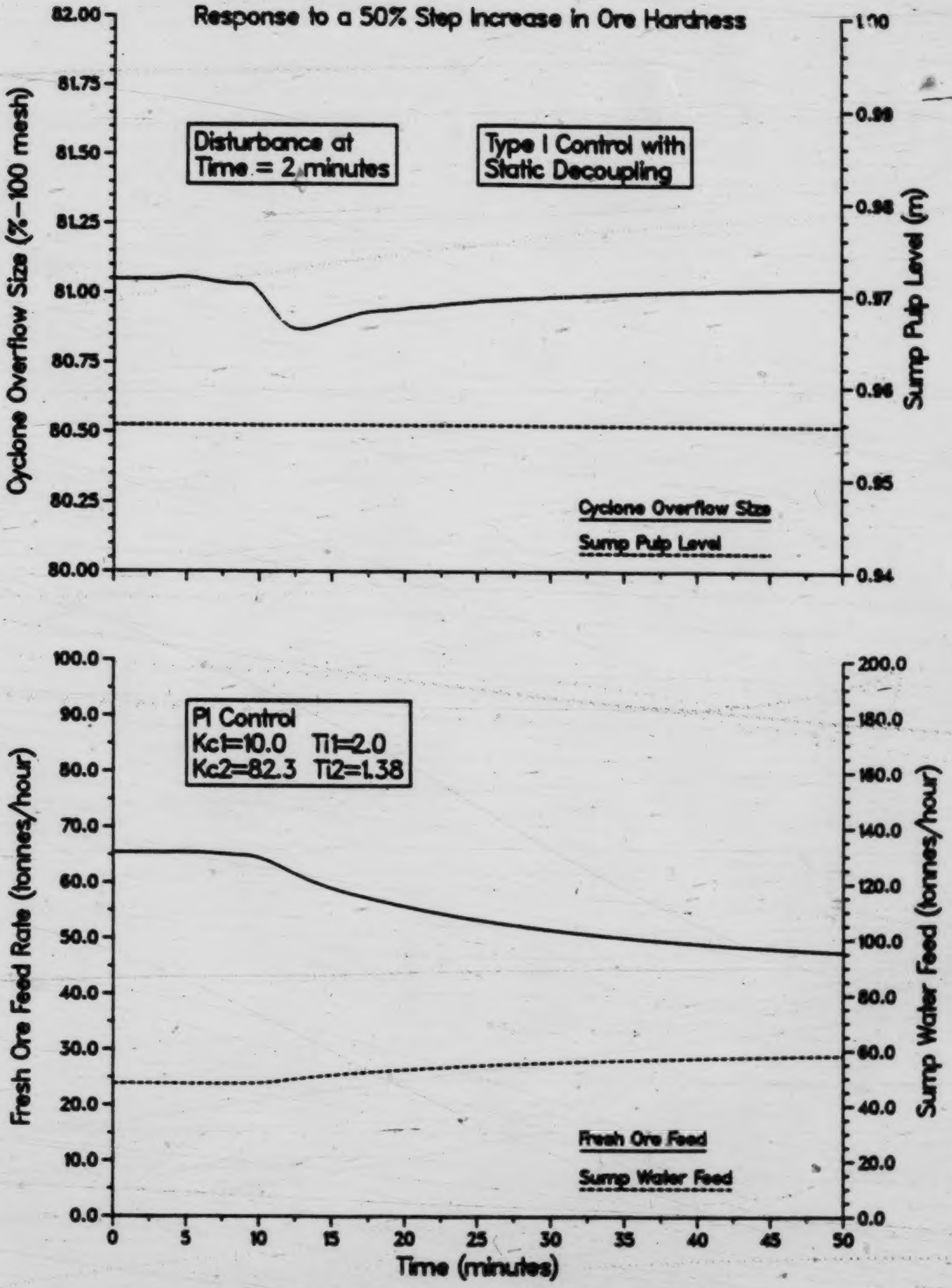


Figure 5.80: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Static Decoupling Controllers

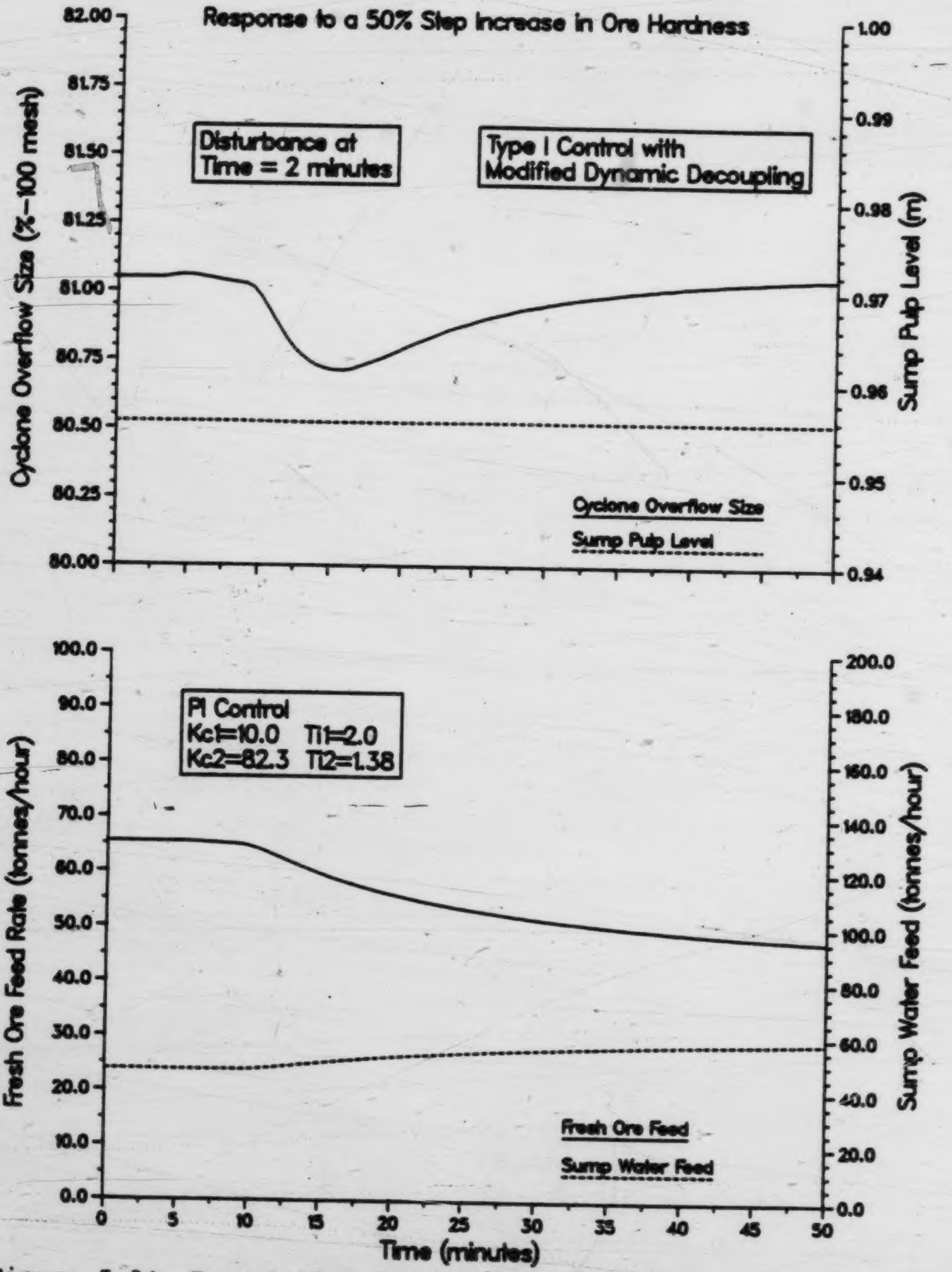


Figure 5.81: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Dynamic Decoupling Controllers

of the control system.

Simulation of the performance of the noninteracting control system for a setpoint change leads to the responses shown in Figures 5.82 and 5.83 for a +3% step increase in the COS setpoint in loop 1. Comparison of the controlled variable responses in Figure 5.82 to those obtained without decoupling action previously shown in Figure 5.35 indicates that the performance has been adversely affected by the inclusion of the static decoupling controllers. However, using dynamic decoupling controllers with a lead time of 1.0 minute for the same setpoint change, produces much more promising results, as evidenced by the responses in Figure 5.83. The response is acceptably quick, with very little overshoot in the COS and the control actions required to produce this response, although somewhat more oscillatory than the interacting case, should still be reasonable in practice. It should be noted that using the original decoupling controllers of equations (5.30) and (5.31) resulted in an unstable responses as was the case with the ore hardness disturbance analysis discussed above. However, unlike the regulatory case, further tuning of the constants in the decoupling controller in loop 1 along with a reduction in the primary controller gain in loop 2, produces some improvement as shown by the response in Figure 5.84.

Table 5.14 summarizes the simulation results for the noninteracting control analysis using the Type I variable pairing scheme.

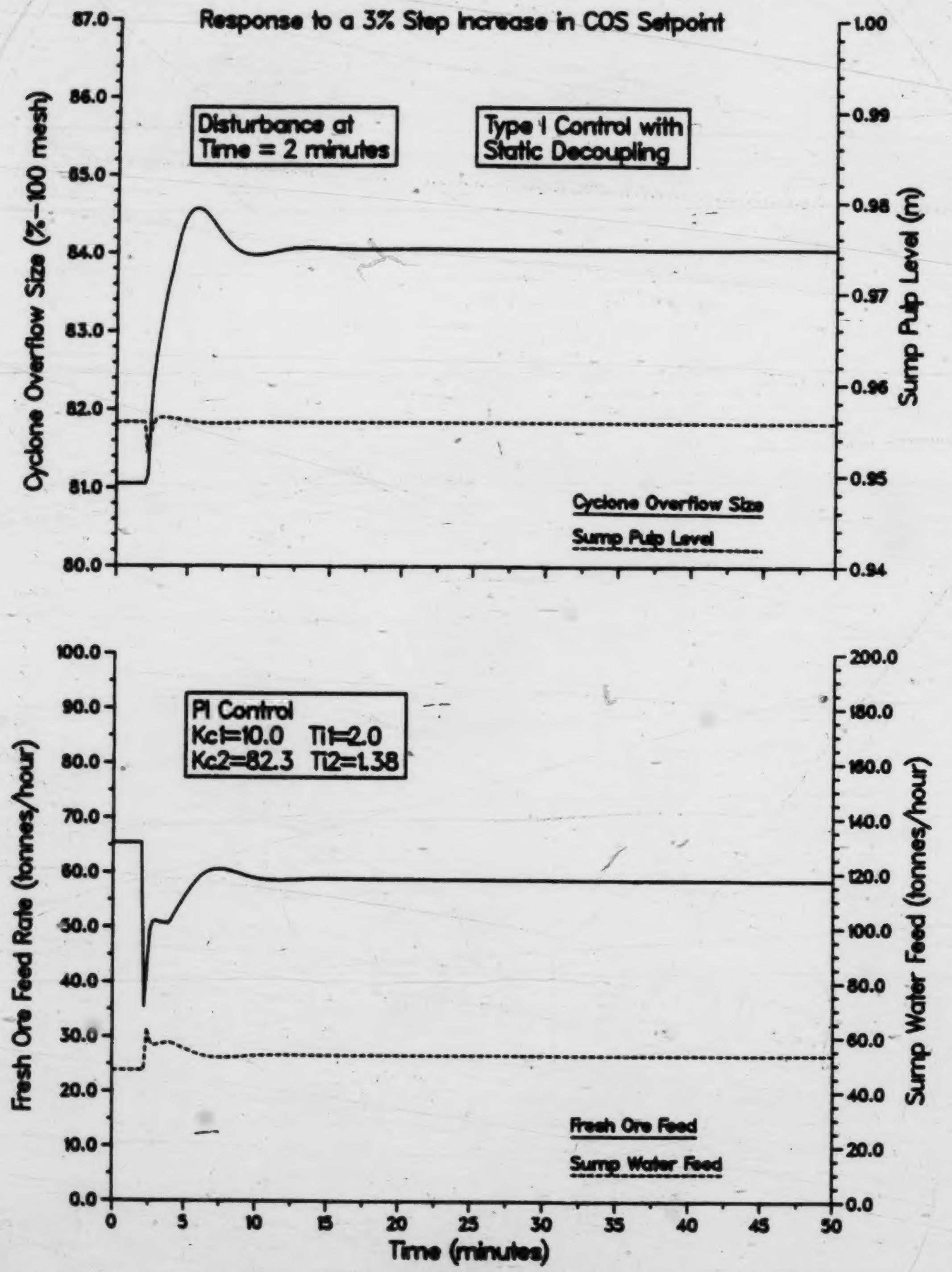


Figure 5.82: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Static Decoupling Controllers

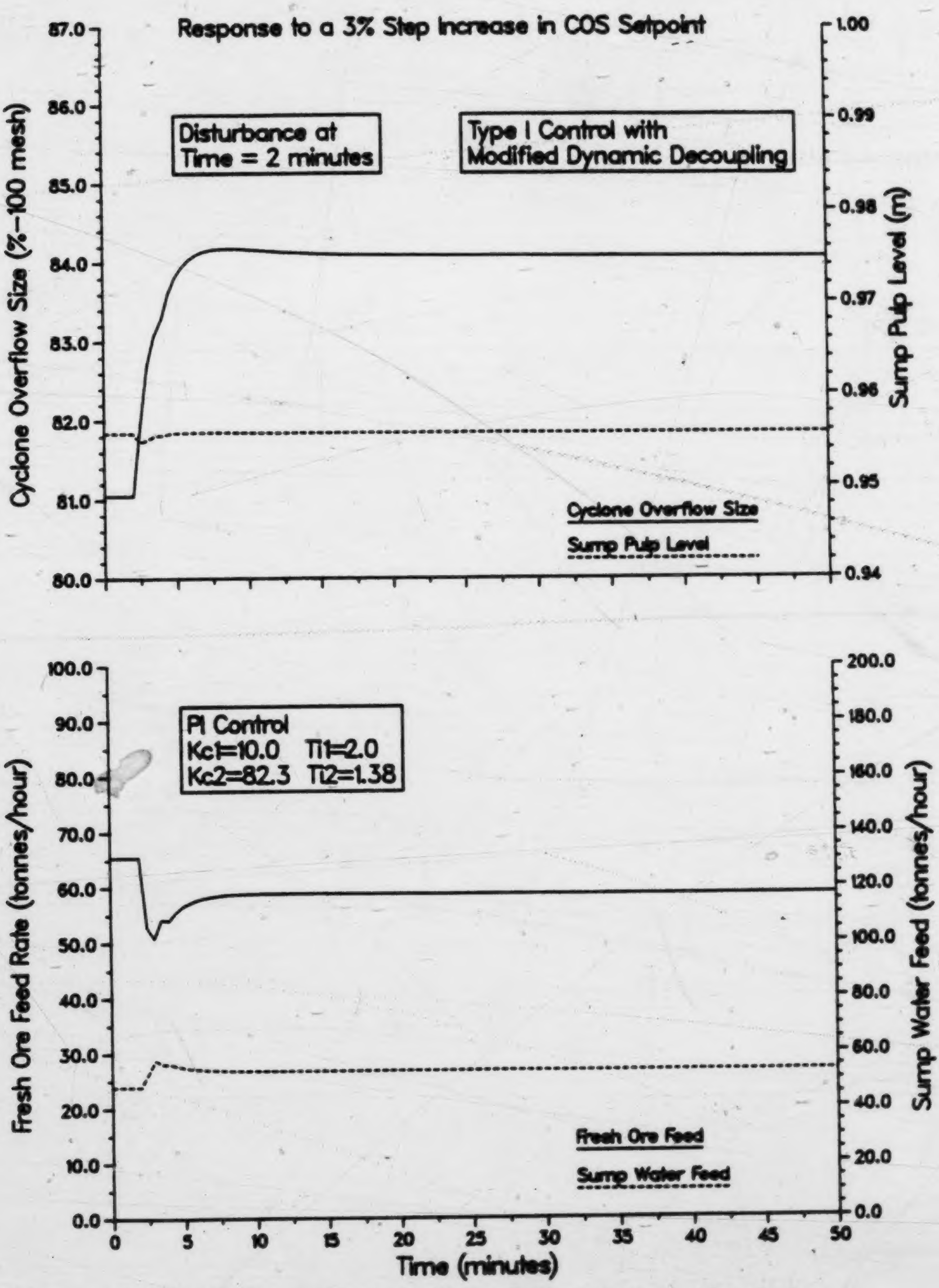


Figure 5.83: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling Controllers

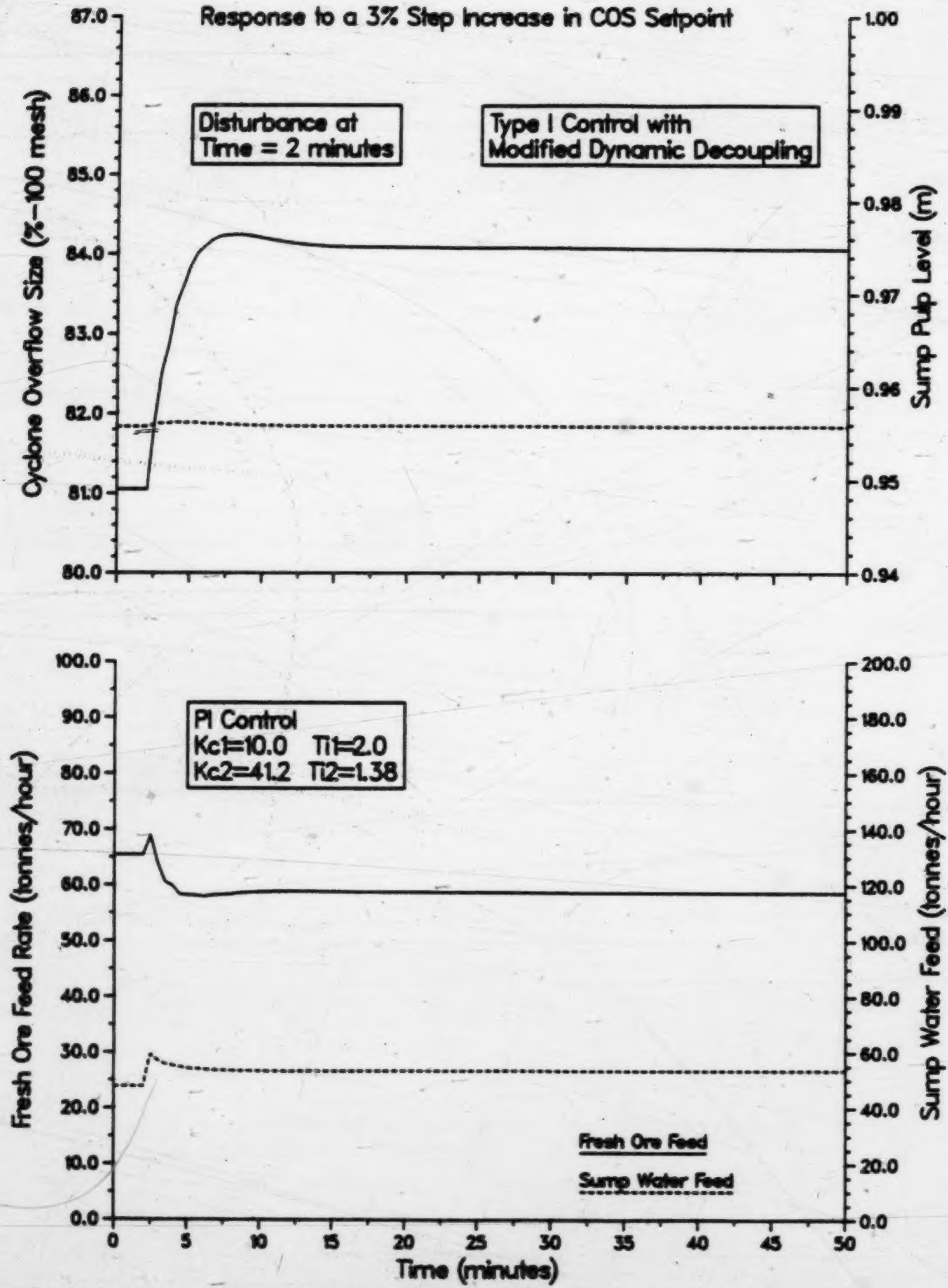


Figure 5.84: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling with Decreased Loop 2 Gain

Table 5.14: Summary of the Simulations for the Lake Dufault Grinding Circuit using Decoupling Control with the Type I Variable Pairing Scheme

Figure Number	Decoupler Type	Comments
5.80	Static	+50% step in hardness. Nonlinear circuit model. No improvement in control performance when compared to Fig. 5.37.
5.81	Dynamic	+50% step in hardness. Nonlinear circuit model. Decouplers detuned to obtain a stable response. Control performance degraded slightly.
5.82	Static	+3% step in COS setpoint. Nonlinear circuit model. Control performance has been adversely affected based on comparison to Fig. 5.35.
5.83	Dynamic	+3% step in COS setpoint. Nonlinear circuit model. Control performance has been improved.
5.84	Dynamic	+3% step in COS setpoint. Nonlinear circuit model. Primary controllers tuned to give better control performance when compared with Fig 5.83.

Simulation of the control behavior that results when using the Type II control scheme leads to results that are similar in nature to those obtained using the Type I control scheme with static decoupling for both hardness and setpoint disturbances. Figure 5.85 shows the results from a simulation for a 50% increase in ore hardness using static decouplers with the gains those of the dynamic decoupling controllers given in Table 5.13. The PI controller constants used were $K_{C1}=10.0$, $\tau_{I1}=2.0$, $K_{C2}=41.2$, and $\tau_{I2}=5.0$. Comparing the responses in this figure with those in Figure 5.50 (Section 5.6), shows that there has been a small improvement in the responses of both the hydrocyclone overflow size and the sump pulp level. However, use of static decoupling for setpoint tracking degrades the performance of the control system as is evidenced by comparing the responses of the controlled variables in Figure 5.86 with those in Figure 5.48.

Simulation of the Type II control system using the dynamic decouplers presented in Section 5.9.2 yields the responses shown in Figures 5.87 and 5.88 for the regulatory and servo control modes, respectively. Unlike the Type I dynamic decouplers, the constants of the decouplers did not require adjustment to produce a stable system response. In fact, improvements in the performance of both control modes are noted when these responses are compared with those obtained without decoupling controllers (cf. Figures 5.48 and 5.50). The most improvement is evident for the setpoint

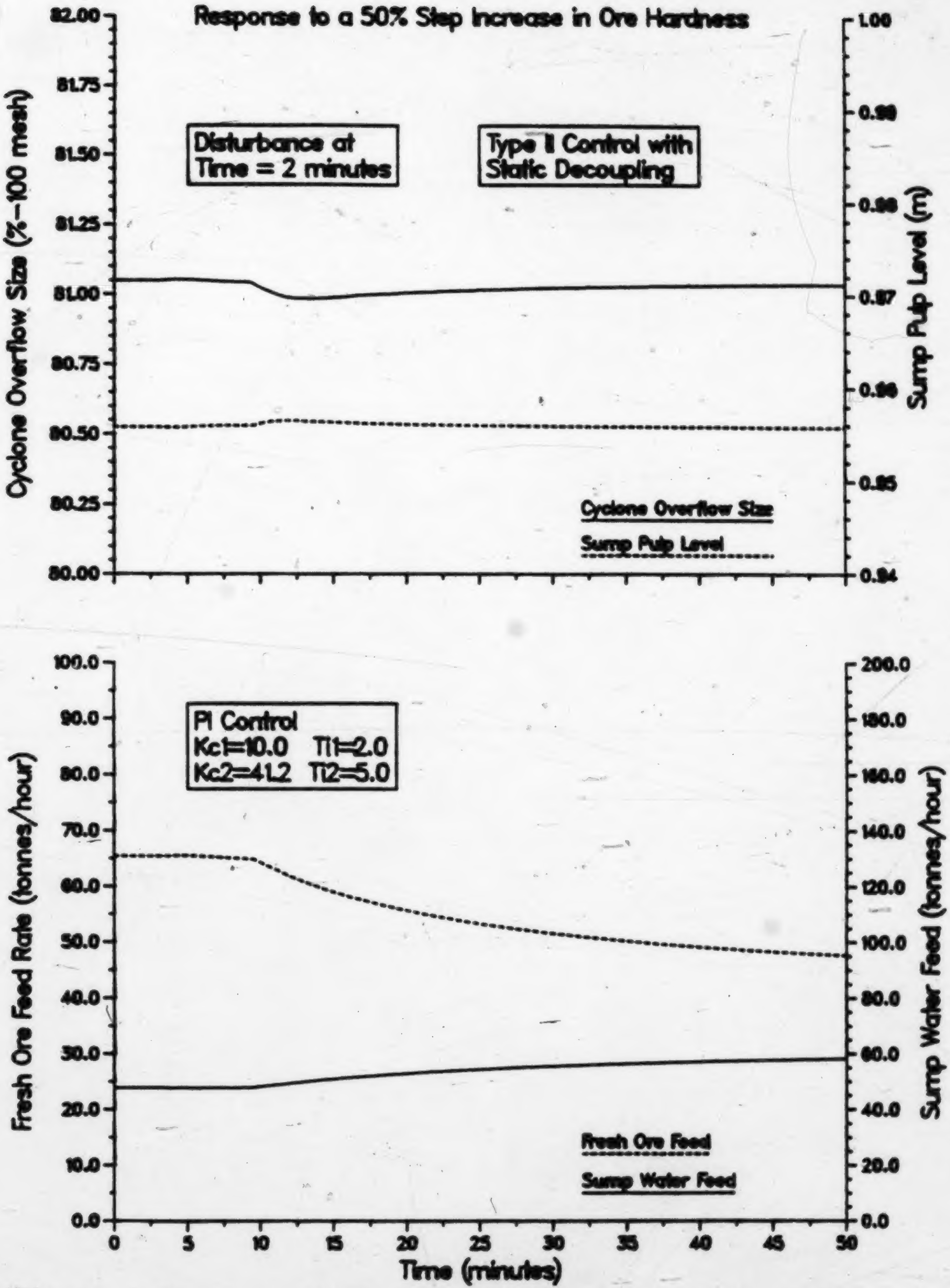


Figure 5.85: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Static Decoupling Controllers

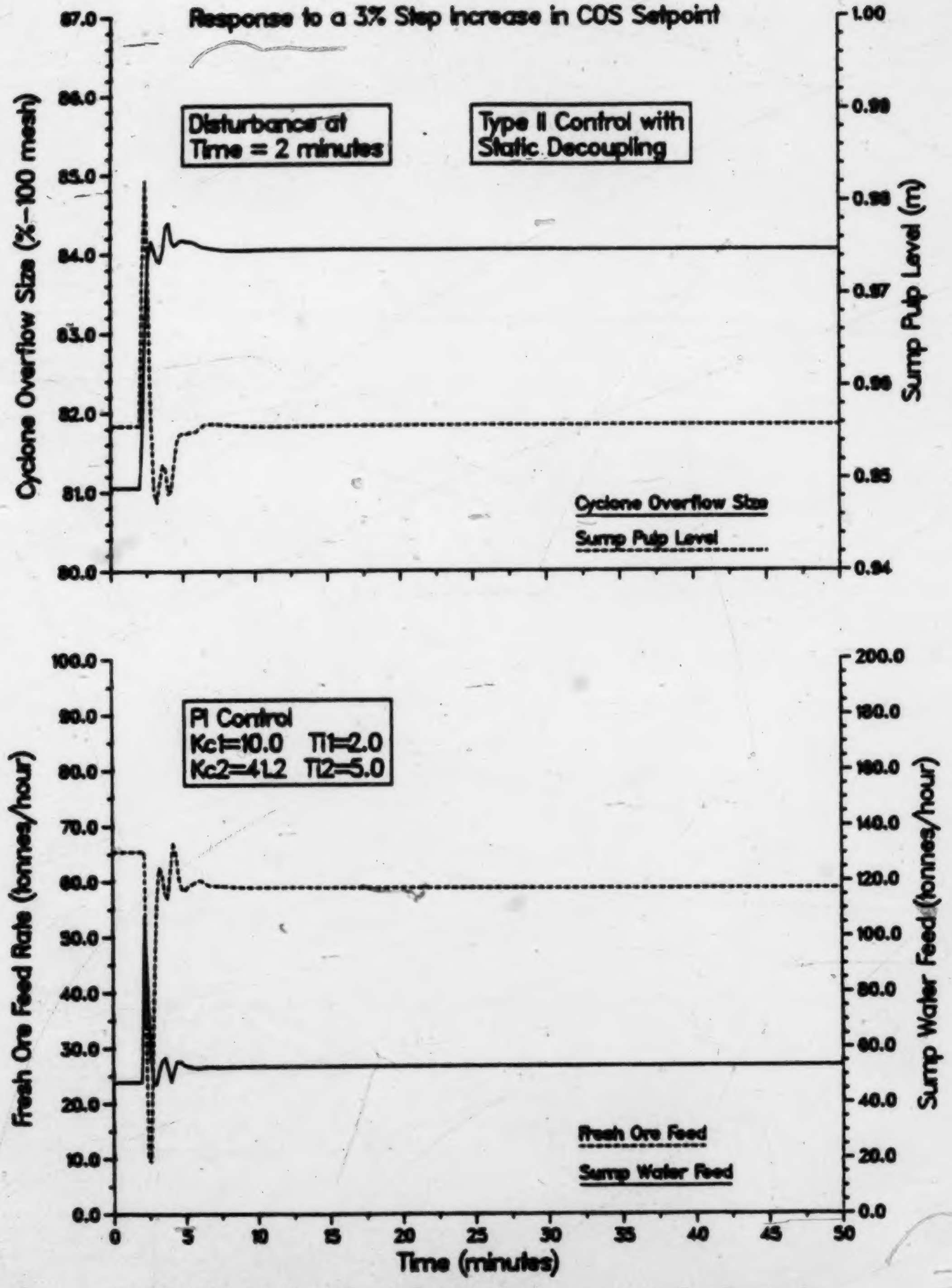


Figure 5.86: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Static Decoupling Controllers

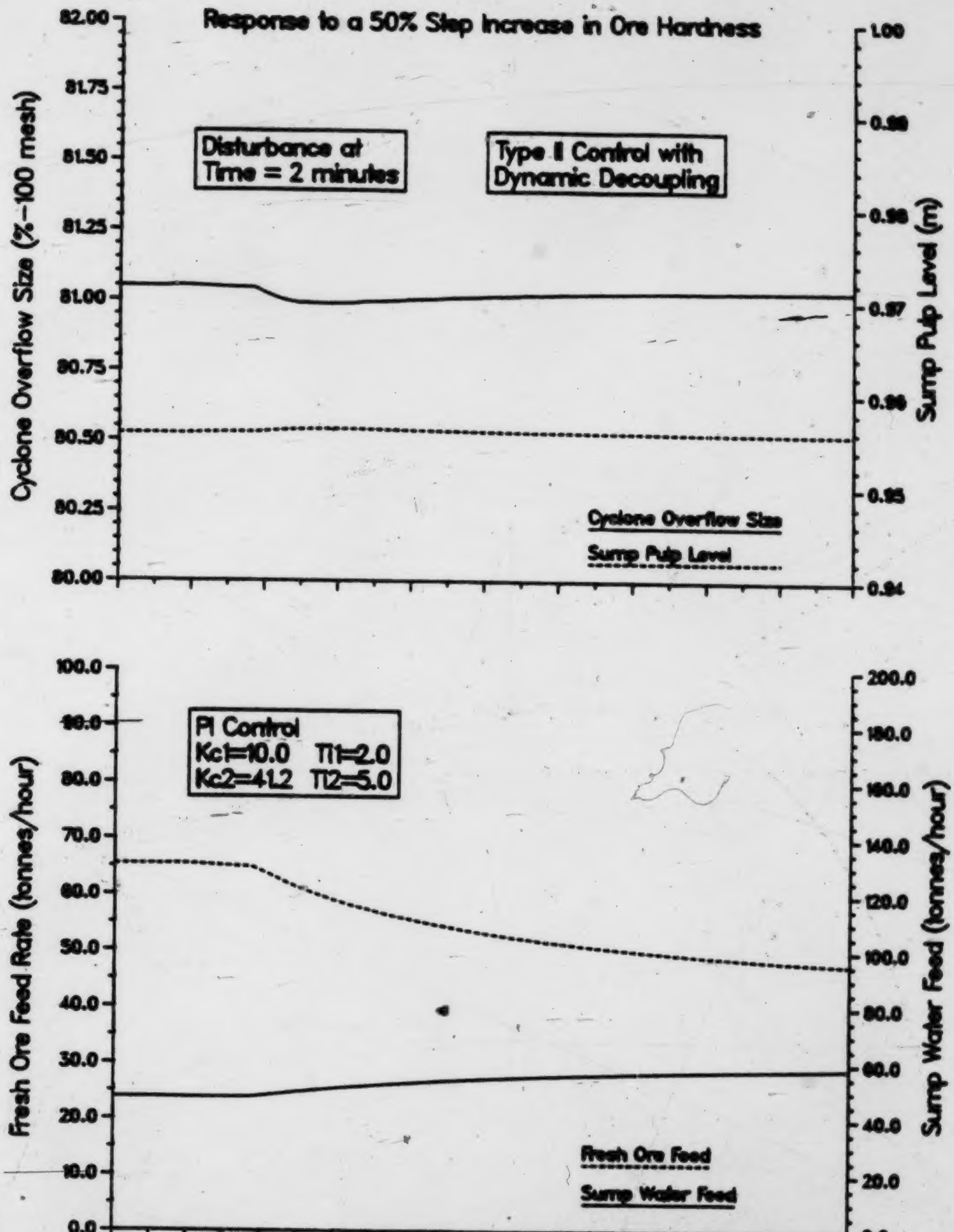


Figure 5.87: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +50% Step in HRD using Dynamic Decoupling Controllers

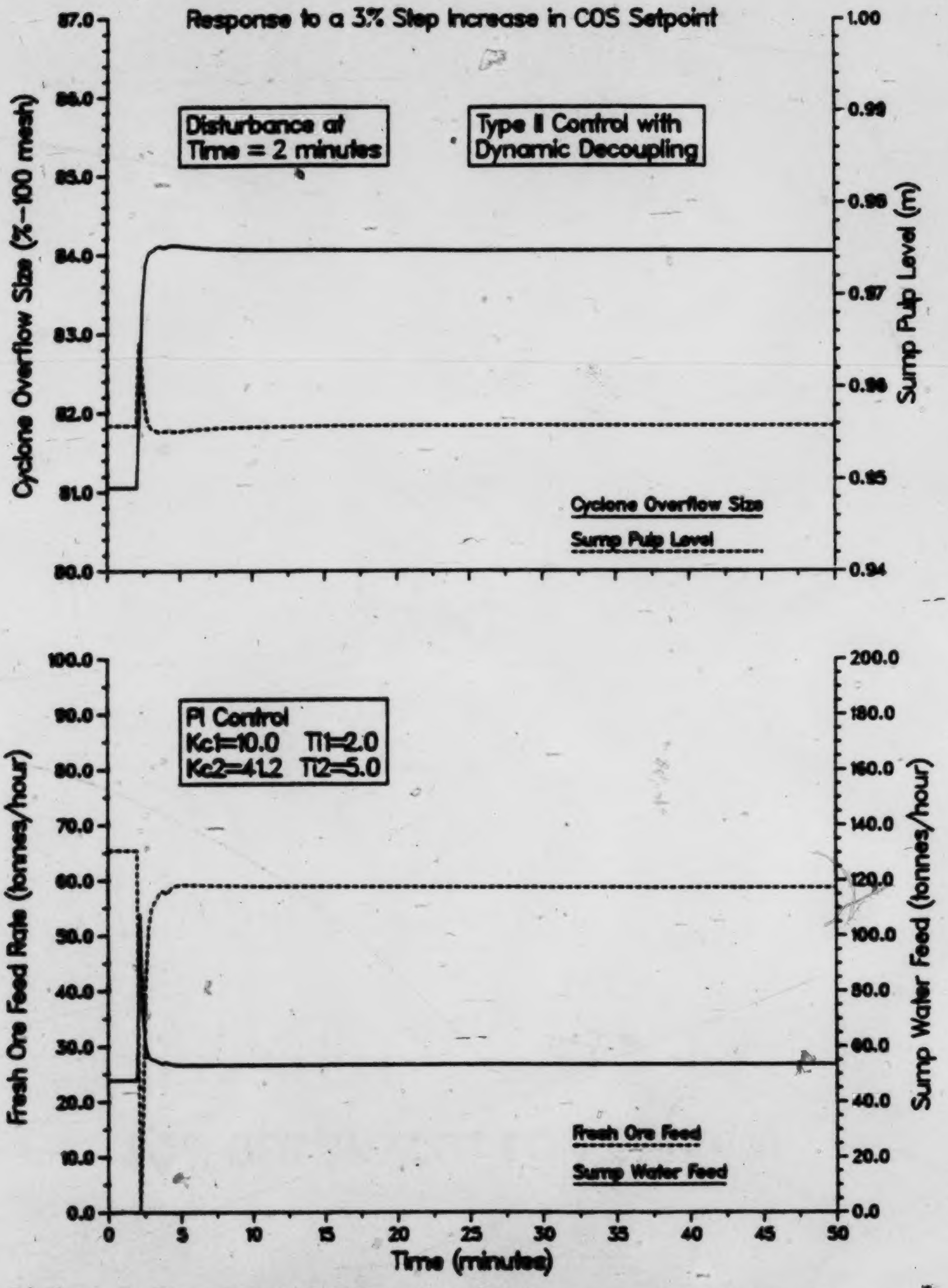


Figure 5.88: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model for a +3% Step in the COS Setpoint using Dynamic Decoupling Controllers

tracking transient for loop 1. However, it should be noted that the required action of the manipulated variable to achieve this response as shown in the bottom plot in the figure is unrealistic. A control action limiter could be used to avoid this problem.

Table 5.15 summarizes the simulation results for the noninteracting control analysis using the Type II variable pairing scheme.

The analysis presented in Appendix E indicates that perfect decoupling of the linear model can be achieved. Simulations for decoupling the linear model confirmed this, but will not be presented for the sake of brevity.

5.11 Adaptive Controller Design

Considering the problems encountered in Section 5.6 with attempting to use the same controller constants to adequately handle both disturbance rejection and setpoint tracking, adaptive control promises to provide a better solution than manually changing the controller constants. The multivariable self-tuning controller described in Chapter 2 (cf. Section 2.7) will be used in this study.

The first step in designing this controller is to choose the sampling times based on the dominant time constant in the system. As suggested by Langman (1987), if τ is the dominant (smallest) time constant then the sampling time should be chosen such that it falls in the interval:

Table 5.15: Summary of the Simulations for the Lake Dufault Grinding Circuit using Decoupling Control with the Type II Variable Pairing Scheme

Figure Number	Decoupler Type	Comments
5.85	Static	+50% step in hardness. Nonlinear circuit model. Small improvement in control performance when compared to Fig. 5.50.
5.86	Static	+3% step in COS setpoint. Nonlinear circuit model. Control performance has been degraded based on comparison to Fig. 5.48.
5.87	Dynamic	+50% step in hardness. Nonlinear circuit model. Decouplers did not require detuning to obtain a stable response. Control performance improved when compared to Fig.5.85.
5.88	Dynamic	+3% step in COS setpoint. Nonlinear circuit model. Control performance has been improved greatly.

$$0.1\tau < T_s < 0.2\tau \qquad (5.32)$$

Because the dominant time constant is independent of the input-output variable pairing scheme, the same sampling times can be used for both the Type I and Type II control system configurations. Using the transfer function model of the grinding circuit (cf. Table 5.5) it can be seen that the dominant time constant for this system is 2.32 minutes. Thus, the practical range for the sampling time is:

$$0.232 < T_s < 0.464 \qquad (5.33)$$

It is convenient to use $T_s=0.3$ minutes as it is an integer multiple of the integration interval, 0.1 minutes.

The procedure for determining the self-tuning controller parameters is discussed in detail by Langman (1987), so will not be repeated here. Using the two by two transfer function model of the process, it can be seen that the order, N, of each loop is three, regardless of which input-output variable pairing scheme is used (i.e. Type I or Type II) because in all cases it is assumed that first order plus time delay transfer functions adequately characterize the process. The time delays of the direct transmission transfer functions in terms of an integer multiple of the sampling time for the Type I configuration are:

$k_1 = 0$

$k_2 = 1$

while for the Type II configuration, the corresponding time delays are:

$k_1 = 1$

$k_2 = 3$

It should be noted that the time delays have been increased in all cases to the nearest integer value. This has the effect of overestimating the time delay in the self-tuning controller which Langman (1987) states as being "best". Using these values along with the values for N given above, it is possible to determine the numbers of coefficients in the controller polynomials in the two by two matricies, G, F, and L. These values are summarized in

Table 5.8. F and L are diagonal matricies. The diagonal elements of G have been subscripted 1 while the off diagonal terms have been subscripted 2, for notational convenience in this table. It should be noted that the H polynomial matrix has not been included because it is related to the process noise which is not employed for the simulations in this work.

The actual values of each of the above parameters is estimated on-line using a recursive least squares estimation involving an upper diagonalization factorization.

The controller weighting parameters are the only remaining variables that require specification. Langman (1987) indicates that typically P and R are set to I, while the form of the Q weighting polynomial is left for the control engineer to choose. There are no set rules or methods for calculating the coefficients of this polynomial. However, Langman (1987) suggests that for a controller structure similar to a PI controller, a reasonable choice for the Q polynomial is:

$$Q = \frac{1 - z^{-1}}{q_0 + q_1 z^{-1}} = \frac{1 - z^{-1}}{K1 + K2z^{-1}} \tag{5.34}$$

where:

$$K1 = K_c[1 + T_s/(2\tau_I)] \tag{5.35}$$

and:

$$K2 = K_c[T_s/(2\tau_I) - 1] \tag{5.36}$$

Hence, the well tuned PI constants used in Section 5.6 can

Table 5.16: Self-Tuning Controller Polynomial Coefficient Numbers for Type I and Type II Configurations

	Type I		Type II	
	Loop 1	Loop 2	Loop 1	Loop 2
n_{G1}	3	4	4	6
n_F	3	3	3	3
n_L	3	4	4	6
n_{G2}	3	4	4	6
Total	12	15	15	21

be used directly to determine the Q weighting polynomial coefficients for each loop. These constants are K_{C1} = -10.0, K_{C2} = 82.3, and τ_{I1} = 2.0, τ_{I2} = 1.38 for the Type I variable pairing, and K_{C1} = -10.0, K_{C2} = 41.2, and τ_{I1} = 2.0, τ_{I2} = 5.0 for the Type II variable pairing. The values calculated for the constants in the Q weighting polynomials are sumarized in Table 5.17.

5.12 Adaptive Control Behaviour

The methodology for using the self-tuning controller is similar to that used by Langman (1987) in his distillation column control studies. Specifically, the identification routine is allowed to operate for a period of time before self-tuning control of the system is actuated. During this "identification stage" discrete PI controllers are used to

Table 5.17: Self-Tuning Controller Q Weighting Polynomial Parameters from the well tuned Continuous PI Controller Constants

Parameter	Type I		Type II	
	Loop 1	Loop 2	Loop 1	Loop 2
K1 or q_0	-10.75	91.21	10.75	42.44
K2 or q_1	9.25	-73.39	-9.25	-39.96

force the system through setpoint transients to allow the identification algorithm to converge on reasonable parameter estimates. The identification algorithm is allowed to operate from the start of the simulation. After a period of about 35 minutes of simulation time, the "self-tuning stage" is started by actuating the self-tuning controller. This controller is then used while the system is subjected to another series of setpoint transients. Finally, the system is subjected to a hardness load disturbance, completing the third stage of the simulation. This proceedure allows the performance of the self-tuning controller to be evaluated under conditions similar to those that were investigated in Section 5.6.

The simulated response of the manipulated and controlled variables of the Type I control scheme with the circuit represented by the nonlinear model is presented in Figure 5.89 and the estimated parameters in Figure 5.90. The square wave setpoint pattern identification phase used to generate these results consisted of a 1.5% step increase in

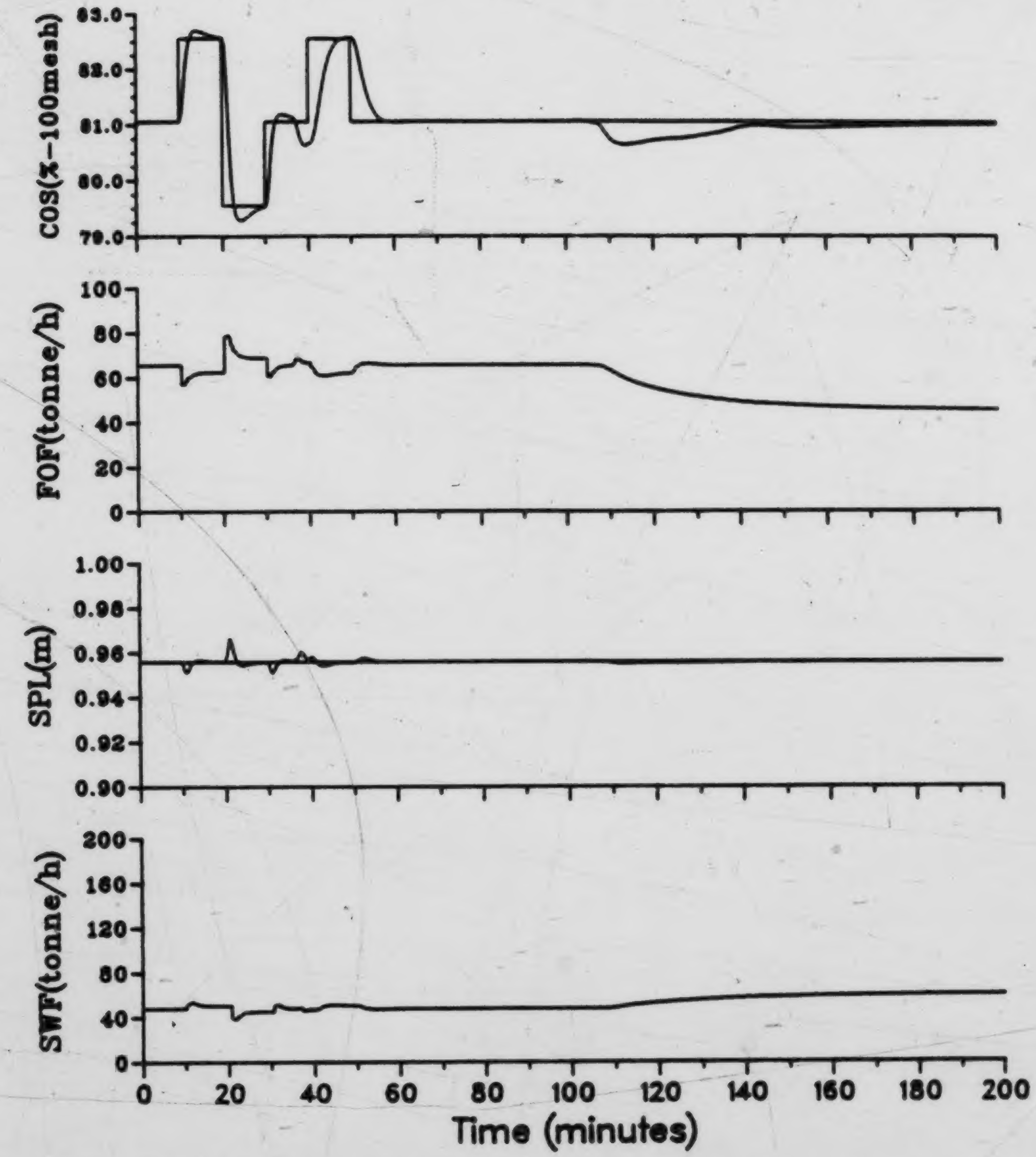


Figure 5.89: Type I Closed Loop Response of the Lake Dufault Nonlinear Circuit Model using Self Tuning Control

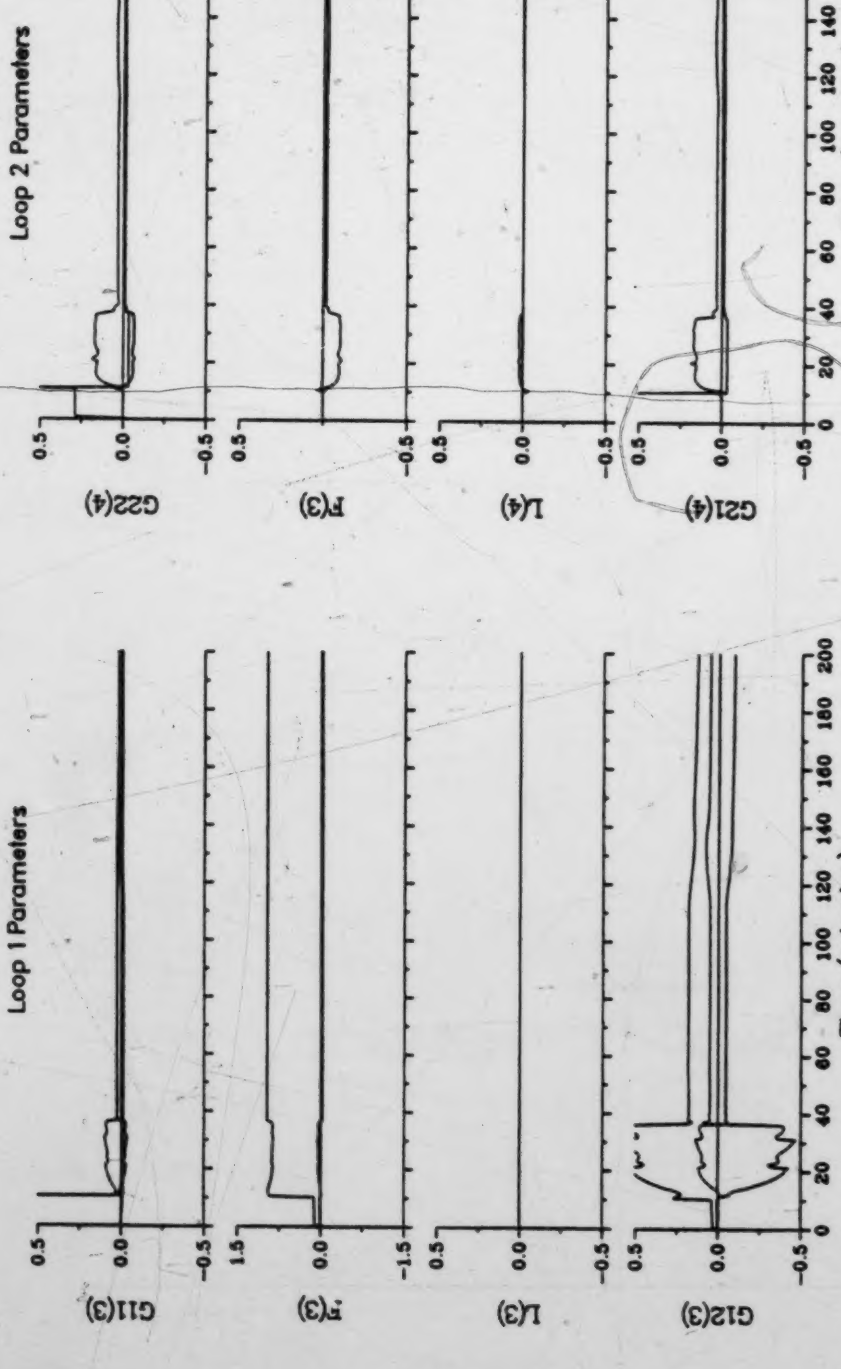


Figure 5.90: Type I Lake Dufault Nonlinear Circuit Model Parameter Estimates

the COS at ten minutes, followed by a 3% step decrease in the COS at twenty minutes. The COS setpoint was returned to its initial steady state value (81.048 %-100 mesh) at 30 minutes and the self-tuning controller was actuated at 35 minutes. At a simulation time of 40 minutes the system was again subjected to a 1.5% step increase in the COS setpoint and at 50 minutes it was returned to the initial steady state once again. A 50% step increase in the ore hardness was introduced at 100 minutes. It should be noted that the discrete nature of the PI controller used for the initial tuning stage necessitated reducing the controller gains to $K_{C1} = 5$ and $K_{C2} = 10$ for loops one and two, respectively, in order to achieve satisfactory control. This also has the effect of changing the Q weighting polynomials to:

$$Q_1 = \frac{1 - z^{-1}}{-5.375 + 4.625z^{-1}}$$

and:

$$Q_2 = \frac{1 - z^{-1}}{11.08 - 8.916z^{-1}}$$

for loops one and two respectively.

It can be seen from Figure 5.89 that the setpoint transient response when the system is under discrete PI control corresponds reasonably well to that found for the continuous control case presented in Figures 5.35 and 5.36. It is evident by examining the COS response that the self-tuning controller provides no improvement in control performance, and in fact, a slight reduction is noted compared to the response under PI control even though the controller model parameters appear to have converged as evidenced by the adaptation patterns shown in Figure 5.90. The load disturbance rejection capabilities of the self-tuning controller are also seen to be similar to the results found using the continuous PI controller that has been detuned for setpoint changes (cf. Figure 5.37).

The behavior of the self-tuning controller using the Type II configuration is similar to that found using the Type I scheme. The same methodology for parameter adaptation was used in this simulation utilizing the same disturbance pattern as described above.

Figures 5.91 and 5.92 present the controlled and manipulated variable response curves and the parameter estimates, respectively. It can be seen from Figure 5.91 that the discrete PI controllers are capable of producing acceptable responses after reducing the controller gains to $K_{C1} = 5$ and $K_{C2} = 5$ for loops one and two respectively, which changes the Q weighting polynomials to:

$$Q_1 = \frac{1 - z^{-1}}{5.375 - 4.625z^{-1}}$$

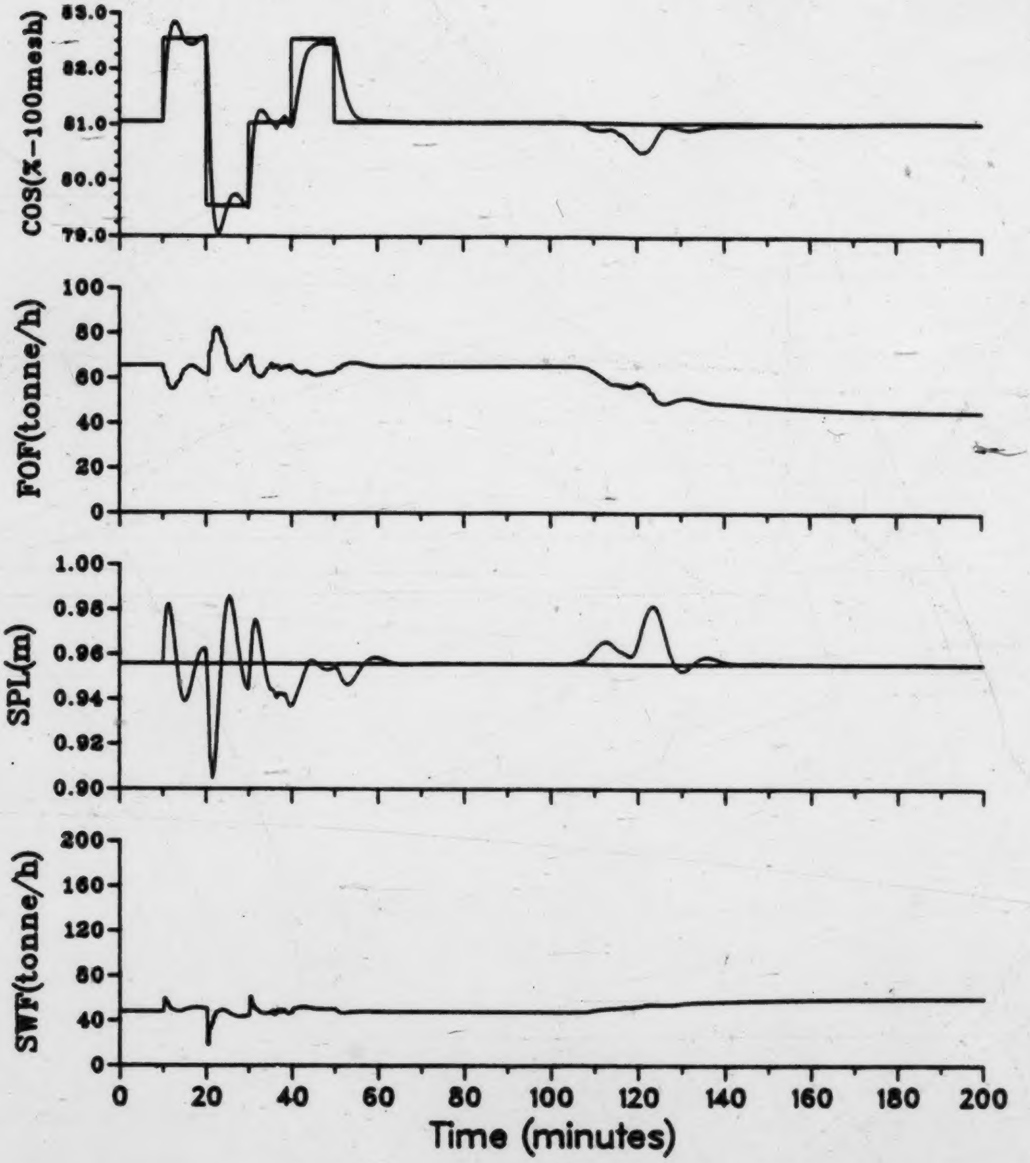


Figure 5.91: Type II Closed Loop Response of the Lake Dufault Nonlinear Circuit Model using Self Tuning Control

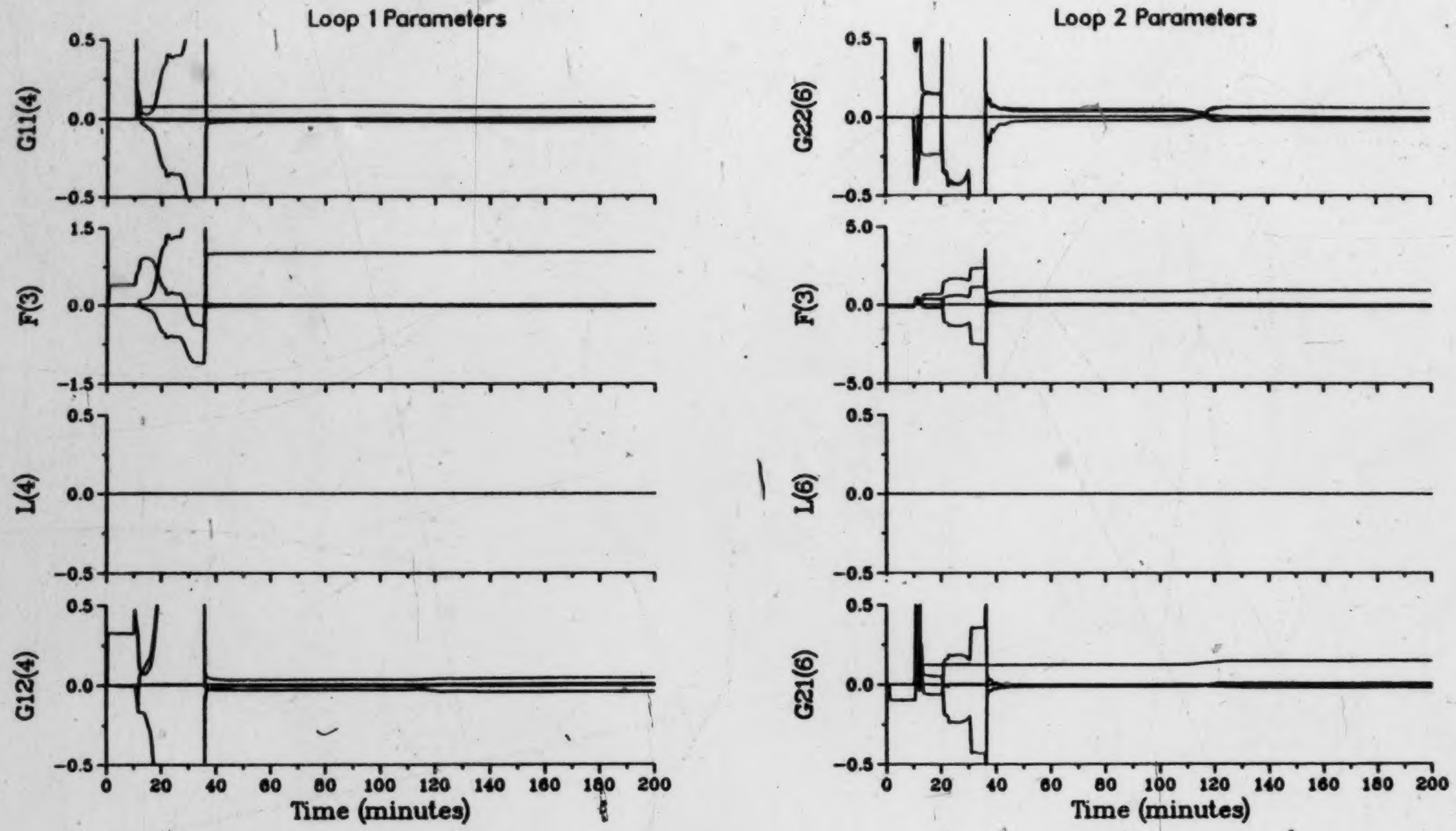


Figure 5.92: Type II Lake Dufault Nonlinear Circuit Model Parameter Estimates

and:

$$Q_2 = \frac{1 - z^{-1}}{5.15 - 4.85z^{-1}}$$

for loops one and two respectively. Again, however, as was the case with the Type I configuration, the self-tuning controller does not perform as well even though the parameters have essentially converged before the self-tuning controller is actuated as evidenced by Figure 5.92. This performance suggests that further tuning of the parameters of the Q weighting polynomial is required. This has been left for future consideration.

6. Digital Control Simulation Study

6.1 Introduction

The purpose of this chapter is to present the results of a discrete versus continuous control system analysis. The grinding circuit used in this study follows that used by Adel et al. (1983) to allow comparison of the results.

The next section begins with a description of the grinding circuit configuration used for this study. Section 6.3 outlines the transfer functions used to model this circuit and Section 6.4 discusses some simulations using continuous control techniques. These results are contrasted to those found using discrete control in Section 6.5.

6.2 Open Circuit Grinding Operation Description

The grinding circuit used for this work is shown schematically in Figure 6.1. It differs from the circuit described in Chapter 5 in that the mill is operated in an open circuit configuration. In other words, there is no separation and recirculation of the oversize product for regrinding. The feed slurry enters the mill through a chute, and is ground in a single pass through the mill, exiting as the product stream.

The control system for a typical open circuit grinding operation is also shown in Figure 6.1. It consists of a particle size analyzer to measure the particle size

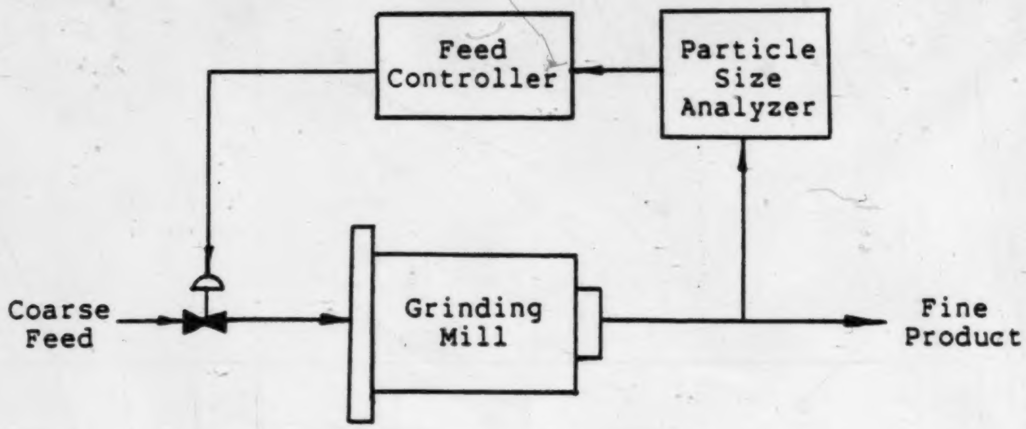


Figure 6.1: Open Circuit Grinding Mill Schematic Diagram

distribution of the product stream, a controller, and a final control element (shown as a valve in the figure).

Following the work of Adel et al. (1983), the controlled variable for this circuit is the specific surface area of the product stream. This is similar in nature to the product size specification used in Chapter 5 where the performance of the grinding circuit was characterized in terms of a mass percent passing a given screen size. The manipulated variable available to control the surface area output is the fresh ore feed flow rate. In the Lake Dufault simulations discussed in Chapter 5, the feed stream particle size distribution was implicitly assumed not to change. In this case however, the feed stream specific surface area is considered as a load disturbance input to the system, along with the grindability of the ore. The grindability is a measure of the hardness of the ore in terms of the power required to increase the surface area of the ore particles through breakage.

6.3 Linear Model

The grinding circuit model used in this study consists of three transfer functions that relate the feed stream surface area, grindability, and flow rate to the product surface area. As outlined by Adel et al. (1983), first and second order transfer functions can be used with reasonable success. The process transfer function that links the product stream specific area (PSA) to the fresh ore feed (FOF) flow rate is:

$$G_p(s) = \frac{PSA}{FOF} = \frac{-0.01}{1s+1} \qquad (6.1)$$

with time in hours. The transfer function relating the product surface area to the feed surface area (FSA) is:

$$G_{L1}(s) = \frac{PSA}{FSA} = \frac{1}{s+1} \qquad (6.2)$$

The second load transfer function ties the feed grindability (GRD) input to the product surface area as follows:

$$G_{L2}(s) = \frac{PSA}{GRD} = \frac{0.001}{s^2+2s+1} \tag{6.3}$$

It should be noted that the parameters used in equations (6.1) to (6.3) correspond directly to those used by Adel et al. (1983) and the numerical values are easily derived from data given by these authors.

The closed loop block diagram presented in Figure 6.2 summarizes the model used in this work. It should be noted that an analysis delay has been included in the form of a time delay in the feedback loop to represent the particle

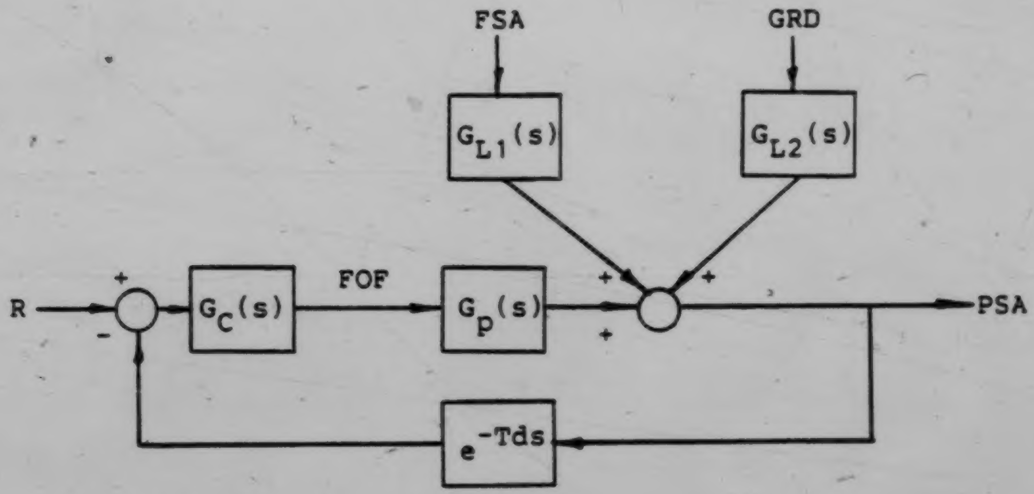


Figure 6.2: Closed Loop Block Diagram for the Open Circuit Grinding Mill Simulation Study

size analyzer. Also, the setpoint comparator used is representitive of the convention used throughout this work, which is opposite to that used by Adel et al. (1983).

Implementation of this block diagram for simulation on a digital computer is easily accomplished using the existing routines in the DYFLO2 (Franks, 1982) subroutine library. The process and load transfer functions are modelled using the routines TFN1 and TFN2, while the analysis time delay makes use of the TDL routine. Proportional, proportional integral, and proportional integral derivative continuous time controllers are also available in the DYFLO2 library.

In order to apply discrete time control strategies to this system, three "samplers" and a zero order hold (ZOH) are incorporated in the original closed loop system as shown in Figure 6.3. An extension to the DYFLO2 control algorithms to include the routine, DISPID, which is capable of simulating discrete time P, PI, or PID controllers following the theory outlined in Chapter 2 (cf. Section 2.3) enables this block diagram to be simulated as well. It should be noted that although a zero order hold routine was also added to the subroutine library, it is inherently included in the implementation of the discrete controller used in this work, and so need not explicitly be used. A more detailed account of the extensions made to the DYFLO2 library can be found in Appendix D.

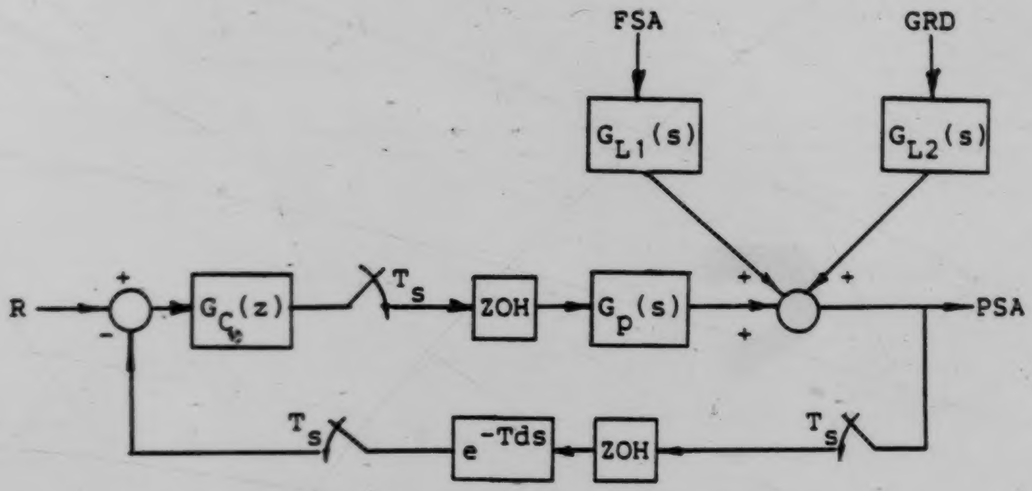


Figure 6.3: Closed Loop Block Diagram for the Open Circuit Grinding Mill Simulation Study with Digital Control

6.4 Continuous Control System Analysis

To enable comparison to the work by Adel et al. (1983), the same steady state conditions and disturbance inputs are used in this study. Table 6.1 summarizes the steady state conditions provided by these authors.

To measure the performance of the control system, a time averaged integral of the absolute value of the error (TIAE) is calculated using:

Table 6.1: Open Circuit Grinding Simulation Equilibrium Conditions

Variable	Steady State Value
Mill Holdup	10,000 kg
Mill Power Input	10 kW
Feed Rate	10,000 kg/hour
Feed Grindability	10,000 m^2/kWh
Feed Surface Area	75 m^2/kg
Product Surface Area	175 m^2/kg

$$\mathrm{TIAE} = \frac{\int_0^t \mathrm{FOF}(\tau)\,|E(\tau)|\,d\tau}{\int_0^t \mathrm{FOF}(\tau)\,d\tau} \qquad (6.4)$$

where:

$$E(t) = \mathrm{PSA}(t) - R \qquad (6.5)$$

The open loop response (i.e. no control) of the grinding circuit to a 22.5 m^2/kg step increase in the FSA combined with a 3000 m^2/kWh step increase in the GRD is presented in Figure 6.4 for comparison to the results presented by Adel et al. (1983). The TIAE for this simulation is 19.15 after 4 hours of simulation time which compares closely to that given by these authors.

To analyze the effect of the particle size analyzer time

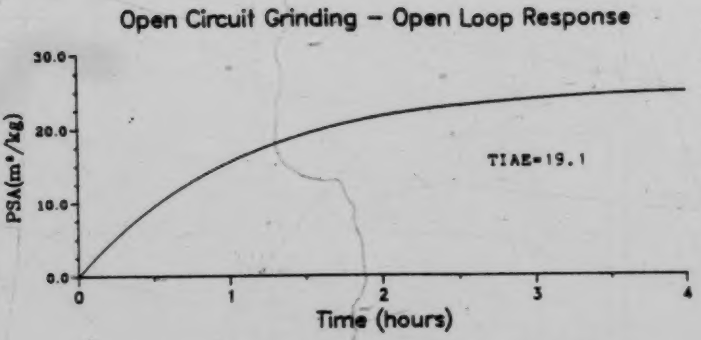


Figure 6.4: Open Loop Response of the Open Circuit Grinding Operation to a Step Disturbance

delay on the stability of the control system, simulations using varying amounts of time delay (T_d) were performed. Simulations using only proportional control with K_C = 400, and time delays ranging from 0 to 0.6 hours produces the responses shown in Figure 6.5. As can be observed, increasing the time delay results in increased oscillation which ultimately results in an unstable system response when T_d = 0.6 hours. With only proportional control there is a steady state offset with TIAE values ranging from 4.37 (T_d = 0) to 8.29 (T_d = 0.6) for four hours of simulation time. Figure 6.6 shows the control behavior that results using PI control (K_C = 400, K_I = 625) with varying amounts of analyzer time delay. The TIAE for the no

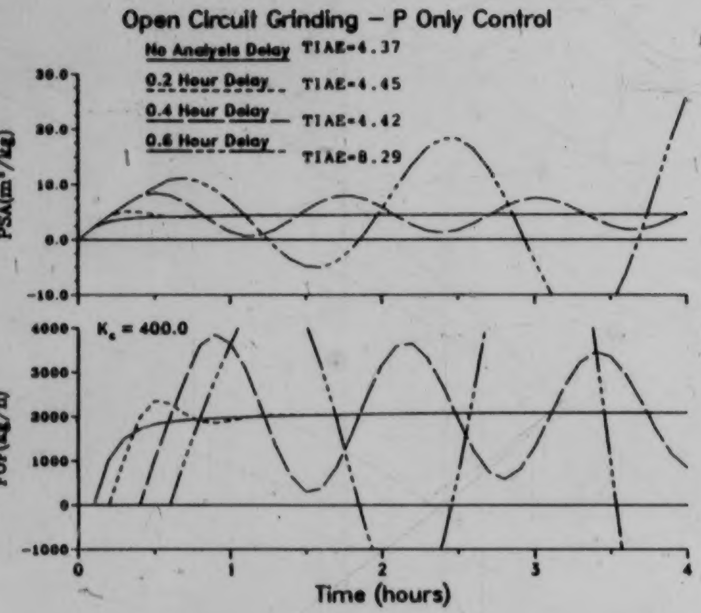


Figure 6.5: Closed Loop Response of the Open Circuit Grinding Operation using Proportion Control to a Step Disturbance and Various Analyzer Time Delays

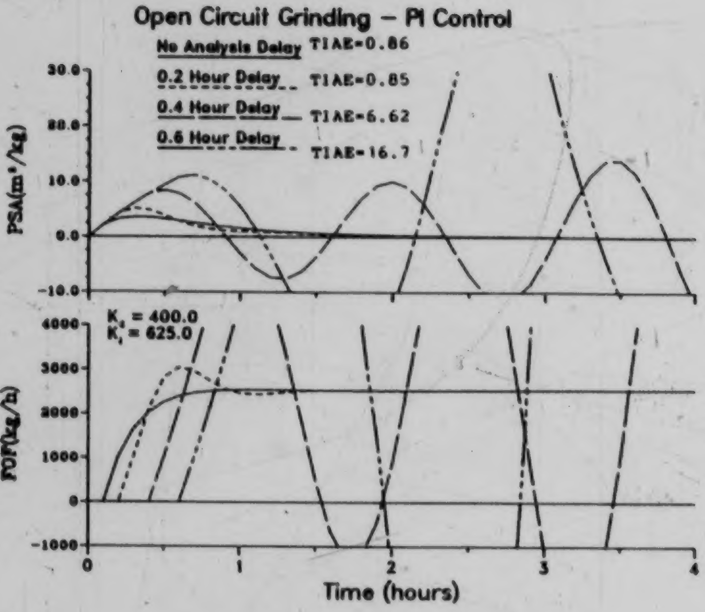


Figure 6.6: Closed Loop Response of the Open Circuit Grinding Operation using Proportion Plus Integral Control to a Step Disturbance and Various Analyzer Time Delays

delay case shows a large improvement in control system performance compared with only proportional control. However, as the time delay is increased to 0.4 hours, no improvement is noted (TIAE = 6.62 with integral action versus TIAE = 4.42 for only proportional action) because of the additional instability introduced into the system by the integral action of the controller. In fact, the system becomes unstable for analyzer time delays of 0.4 hours and longer for these PI controller settings. Time delays play a key role in the performance of control systems so the controller constants must be tuned to obtain satisfactory behavior.

Adel et al. (1983) suggest using the Ziegler Nichols quarter decay ratio tuning fromula for calculating the "best" controller constants for systems with time delays. These formulae are summarized in Table 6.2 and were adapted from those given by Ziegler and Nichols (1942) for consistency with the nomencalture used in this chapter.

Utilizing these formulae and the process transfer function given by equation (6.1) with an analyzer time delay of 1.0 hour gives the controller constants summarized in Table 6.3.

The control behavior that results using these three different controllers is shown in Figure 6.7. The disturbances used to produce these results were a 22.5 m^2/kg step increase in the FSA combined with a 3000 m^2/kWh step increase in the GRD. It can be seen from this figure that

Table 6.2: Ziegler Nichols Quarter Decay Ratio Tuning Formulae

Controller	K_C	K_I	K_D
P	$t/(KT_d)$	-	-
PI	$0.9t/(KT_d)$	$K_C/(3.33T_d)$	-
PID	$1.2t/(KT_d)$	$K_C/(2.0T_d)$	$K_C T_d/2$

the PID controller produces the best overall response, as expected. These responses result in TIAE performance indices of 11.70 for proportional control and a TIAE value of 10.29 and 6.87 for PI and PID control, respectively. The addition of derivative action provides additional stability which allows both the proportional and integral gains to be increased, accounting for the performance improvement.

It is interesting to note that the rate amplitude used in the PID controller has a large effect on the stability of the system. The PID response in Figure 6.7 was obtained using a PID controller with a rate amplitude (a) of 0.33, which is well beyond the 0.05 to 0.10 range suggested by Smith and Corripio (1985). Reduction of the rate amplitude to 0.1 gives rise to the response shown in Figure 6.8 with any further reduction resulting in unstable behavior for the controller settings given in Table 6.3. The responses in Figures 6.7 and 6.8 clearly show that the response using the larger rate amplitude is more desirable which is substantiated by TIAE values of 6.87 and 7.01 for a = 0.33

Table 6.3: Ziegler Nichols Controller Constants

Controller	K_C	K_I	K_D
P	100.0	-	-
PI	90.0	27.0	-
PID	120.0	60.0	60.0

and 0.1, respectively. Explanation of these results may come through numerical roundoff considerations and the fact that first order Euler integration was used for all simulations presented in this work. Alternately, the derivative constant could have been reduced, with similar results to be expected.

6.5 Discrete Control System Analysis

Discrete time control of a grinding circuit employing a particle size analyzer as the sampling device comes naturally through the realization that this analysis is not a continuous operation, as implied in the last section. Since the particle size analyzer samples the process output only at specific instances in time, a time delay arises from the fact that it takes some time for the analyzer to perform its function. New product stream composition information is known only at discrete intervals of time which correspond to the sampling time plus the anlyzer time delay. Thus it is not possible to take control

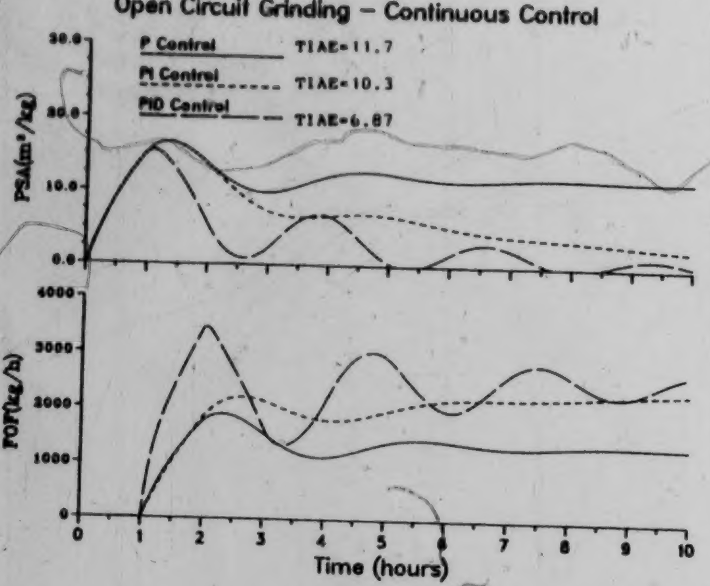


Figure 6.7: Closed Loop Response of the Open Circuit Grinding Operation using P, PI, and PID Control to a Step Disturbance and a 1 Hours Analyzer Time Delay

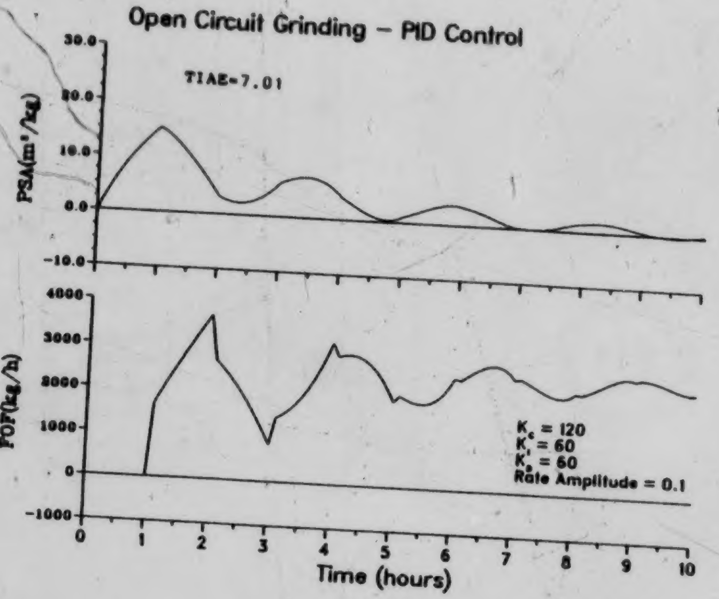


Figure 6.8: Closed Loop Response of the Open Circuit Grinding Operation using a PID Controller with a Rate Amplitude of 0.1

action in the intervening time.

Isermann (1981) states that if the sampling time is small enough, the PID settings for the continuous time controller may be used in the discrete version of the controller with reasonable success. Figure 6.9 shows the simulated responses using sampling times of 0.25 and 0.5 hours with a one hour analyzer delay for the system subjected to the same step input disturbance used for the continuous time simulations discussed in the previous section. Included in this figure is the response of the grinding circuit under continuous PID control. Comparing the TIAE performance index in each case, it is found that

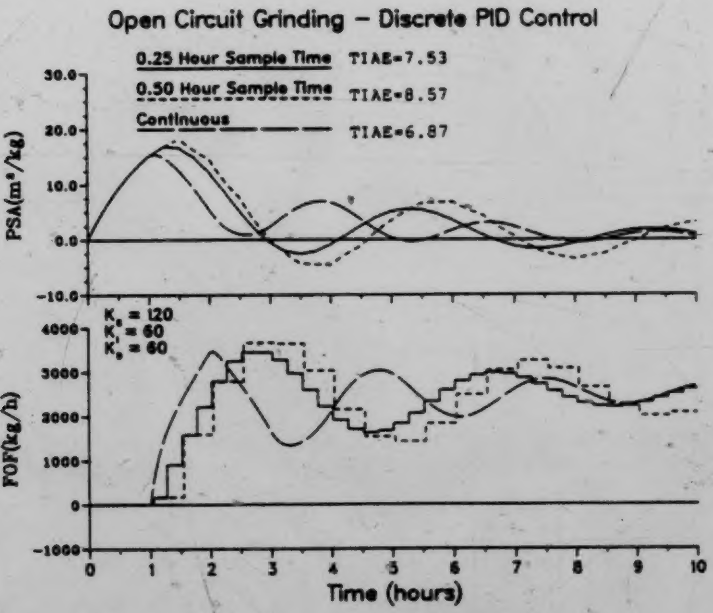


Figure 6.9: Closed Loop Response of the Open Circuit Grinding Operation using Discrete PID Control with Various Sample Intervals

continuous control provides marginally better performance than the 0.25 hours sampling case (TIAE = 6.87 versus TIAE = 7.52). Performance degrades rapidly as the sampling time is increased to 0.5 hours resulting in a TIAE value of 8.57. It should be noted here that the timing of the sampling with respect to both the collection of product samples and calculation of appropriate control action is important. The product sampler (cf. Figure 6.3) should operate slightly before the analyzer and controller output samplers, which close simultaneously. The length of the offset between sampler operation is proportional to the time required for the discrete controller to compute and output the appropriate control action. This offset is small, usually in the order of fractions of seconds, and will be ignored for purposes of this work. Incorrect sampler operation can lead to the response shown in Figure 6.10, where the control action is further delayed by one sampling interval. It is interesting to note that this situation will also arise when the disturbance enters the system at time 0+ (shortly after a sample is taken) as opposed to 0- (shortly before a sample is taken). Figure 6.11 illustrates this point with the sampling/control period represented by vertical bars, while the arrows pointing upward indicate a disturbance input to the system and the arrows pointing down show when the controller will sense the necessity for control action changes. Clearly, the system response shown in Figure 6.10 demonstrates the worst case scenario where

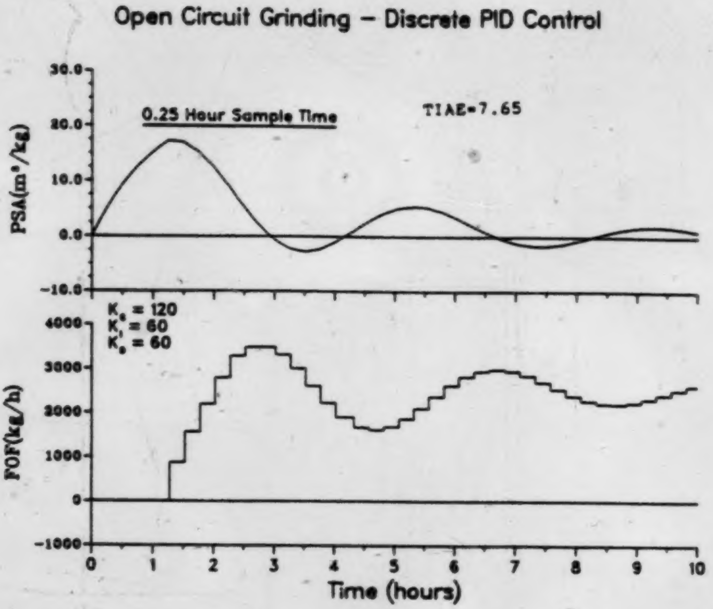


Figure 6.10: Closed Loop Response of the Open Circuit Grinding Operation using Discrete PID Control with Incorrect Sampler Operation

the disturbance enters the system a short time after a product sample is taken. The controller does not take control action until one sampling time after new product information is available. The performance index, when using a sampling time of 0.25 hours is 7.65 so it appears that the performance of the control system has been degraded slightly when in fact it is due to the time of the disturbance. The remainder of this work will assume that the disturbance always enters the system at time 0-.

A pulse input to the system with a frequency of 0.125 hours, as shown in Figure 6.12, will provide a more realistic test of the performance of the control

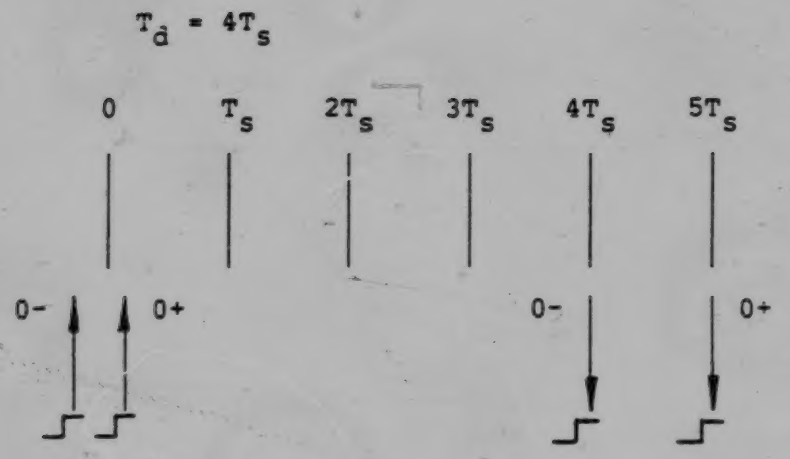


Figure 6.11: Schematic Diagram Showing the Disturbance Input Relative to Sampler Operation

system. The pulse consists of a series of step changes in the feed stream particle surface area and grindability with absolute magnitudes of 22.5 m^2/kg and 3000 m^2/kWh, respectively, with a total duration of 100 hours. The open loop response of the grinding circuit to this pulse input is shown in Figure 6.13, and the TIAE for this simulation is 14.09 after 100 hours. Both the surface area time trajectory and the performance index agree well with results presented by Adel et al. (1983).

It was mentioned earlier that if the sample time was small enough, the Ziegler Nichols tuning formulae presented in Table 6.2 can be used to determine appropriate controller settings for the discrete controllers. However, Adel et al.

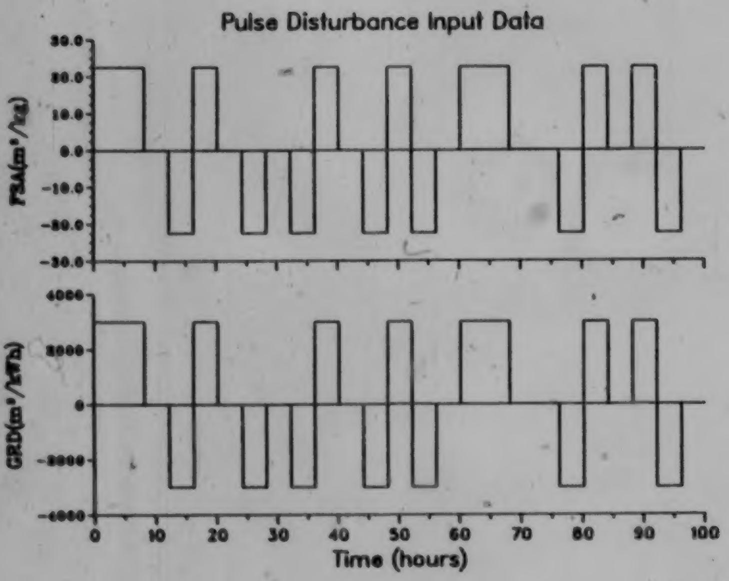


Figure 6.12: Pulse Disturbance Input Signal

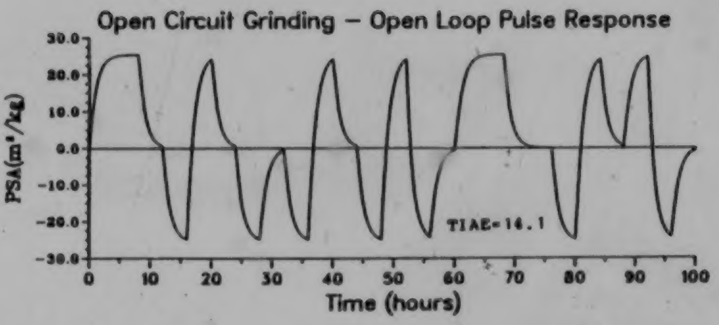


Figure 6.13: Open Loop Response of the Open Circuit Grinding Operation to a Pulse Disturbance

(1983) suggests using a set of modified tuning formulae which take into account the effect of sample time. These formulae, adapted for consistency with the nomenclature used in this work, are summarized in Table 6.4.

Adel et al. (1983) state that an automated particle size analysis system can be expected to have an analysis time delay of 0.125 hours. Using this time delay, along with the appropriate constants from equation (6.1), the controller constants for various sampling times can be easily calculated. The results of such calculations for sampling times of 0.25 and 4.0 hours are summarized in Table 6.5. The choice of these two sampling times stems from consideration of the Shannon sampling theorem which states that the sampling frequency should be less than half the highest frequency component of the sampled signal to enable adequate reconstruction of this signal (Phillips and Nagle, 1984). In other words:

$$1/T_s < 2f_d \qquad (6.6)$$

where f_d is the dominant frequency component of the sampled signal. In this case where the load input pulse train has a frequency of 1/8 hour^{-1}, a sampling time no greater than 4 hours should be used.

Figure 6.14 presents the response of the open circuit grinding operation to the pulse load input using only proportional control action for a sample time of 0.25 hours for a controller gain of 267.0 (cf. Table 6.5). It is clear

Table 6.4: Modified Ziegler Nichols Digital Controller Tuning Formulae after Adel et al. (1983)

Controller	K_C	K_I	K_D
P	$\frac{t}{K(T_d+T_s)}$	-	-
PI	$\frac{0.9t}{K(T_d+T_s)} - \frac{K_I}{2}$	$\frac{0.27tT_s}{K(T_d+T_s)^2}$	-
PID	$\frac{1.2t}{K(T_d+T_s)} - \frac{K_I}{2}$	$\frac{0.6tT_s}{K(T_d+.5T_s)^2}$	$\frac{0.6t}{KT_d}$

from this figure that although the control does provide a great improvement over the open loop case, TIAE = 4.03 versus TIAE = 14.09, a steady state offset occurs for each pulse disturbance. Figure 6.15 shows the system response under PI control using the Ziegler Nichols controller constants given in Table 6.5 with a 0.25 hour sampling rate. Although PI control attempts to remove the steady state offset, the performance index is 4.39, which indicates that the overall performance has been reduced slightly over the P only case. This is misleading because the duration of the pulses is generally to short for the PI controller to return the system to steady state before the next pulse enters the system. It should be evident from the step response results presented in Section 6.4, that proportional plus integral control would perform better than proportional control if longer duration pulses were used. In addition,

Table 6.5: Modified Ziegler Nichols Digital Controller Constants

Controller	Sample Time (hours)	K_C	K_I	K_D
P	0.25	267.0	-	-
	4.0	24.0	-	-
PI	0.25	186.0	108.0	-
	4.0	9.9	24.0	-
PID	0.25	200.0	240.0	240.0
	4.0	2.5	53.0	15.0

the PI controller increases the oscillatory behavior of the closed loop system and over a longer period of time than used here, the removal of the steady state offset can be expected to overcome the loss incurred by this behavior. It is interesting to note that simply tuning the PI controller can improve the performance as illustrated by Figure 6.16. The controller gain was increased to 250.0 to obtain this response and the TIAE for this simulation was 3.80 after 100 hours.

Further preformance improvements can be realized by the addition of a derivative mode to the controller. Figure 6.17 presents the response of the grinding circuit to the pulse train using a digital PID controller with a 0.25 hour

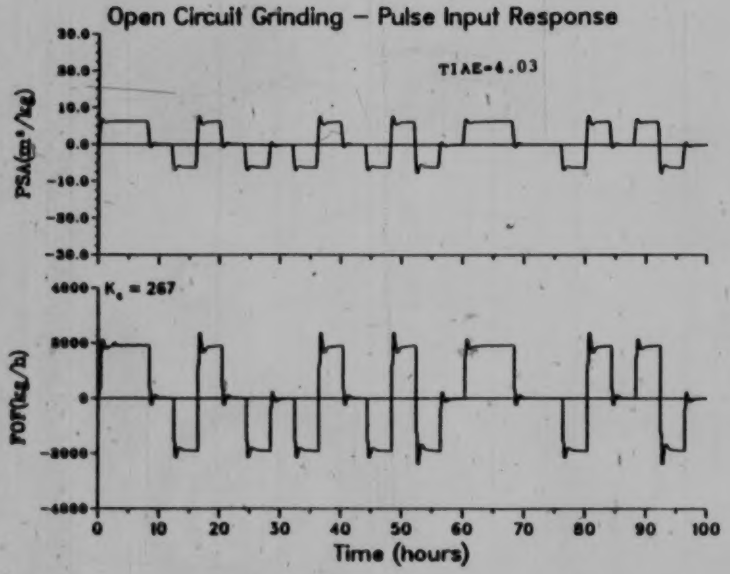


Figure 6.14: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete Proportion Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance

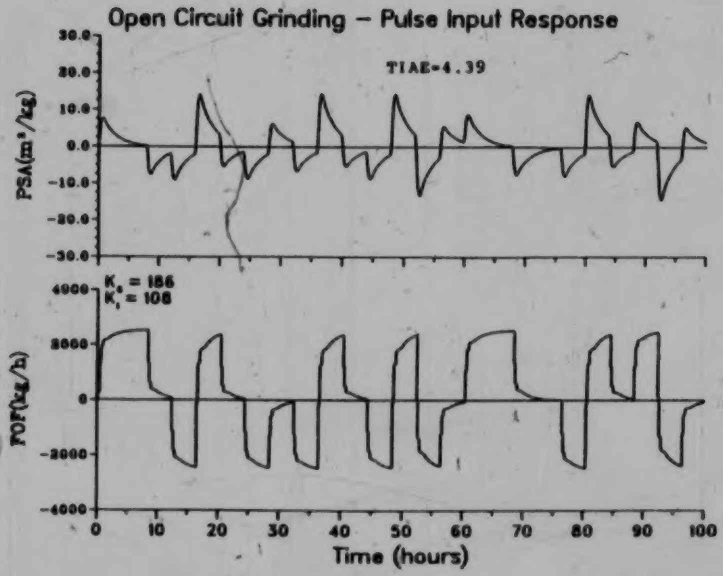


Figure 6.15: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance

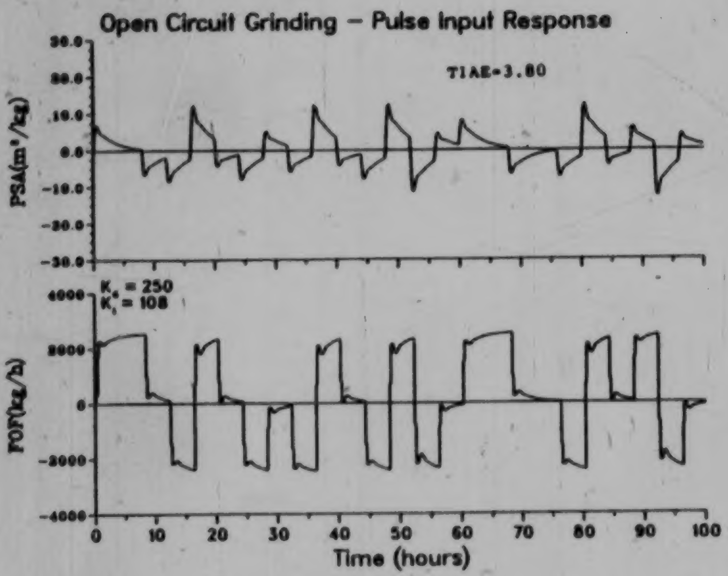


Figure 6.16: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with Tuned Controller Constants for a Pulse Disturbance

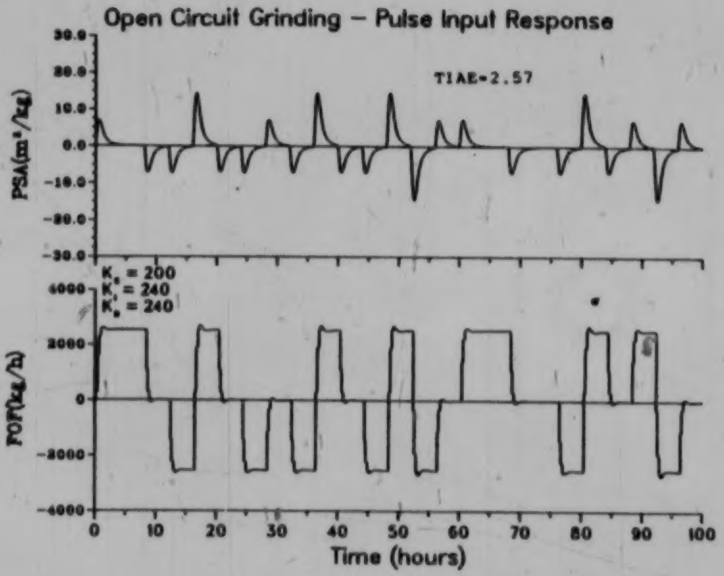


Figure 6.17: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PID Controller with a Sample Time of 0.25 Hours for a Pulse Disturbance

sampling rate and the PID controller constants in Table 6.5. The performance index for this simulation was 2.57, which clearly indicates the improvement over the P or PI control cases discussed earlier.

To demonstrate the effect of using a sampling time that is too long, results of simulations using all three types of controllers and a sampling rate of 4.0 hours are presented in Figures 6.18 through 6.20. The controller constant used for these simulations are summarized in Table 6.5 and were estimated using the Ziegler Nichols digital controller tuning formulae given in Table 6.4. It is clear from these figures that proportional only control (cf. Figure 6.18)

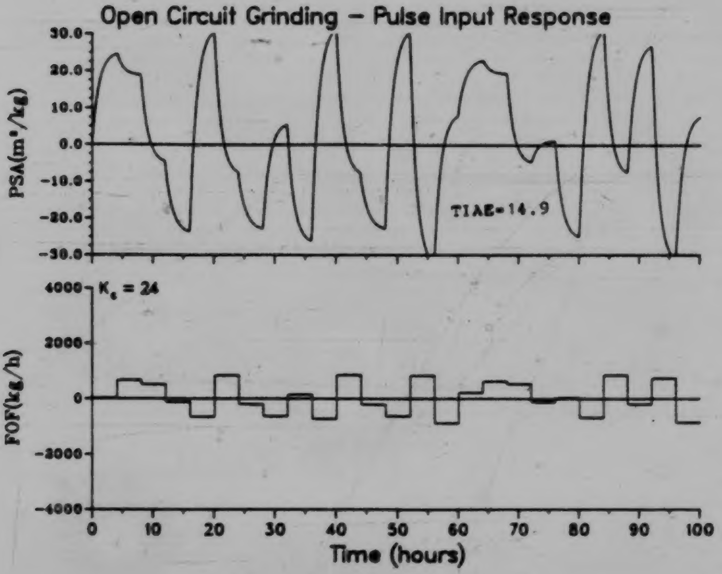


Figure 6.18: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete P Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance

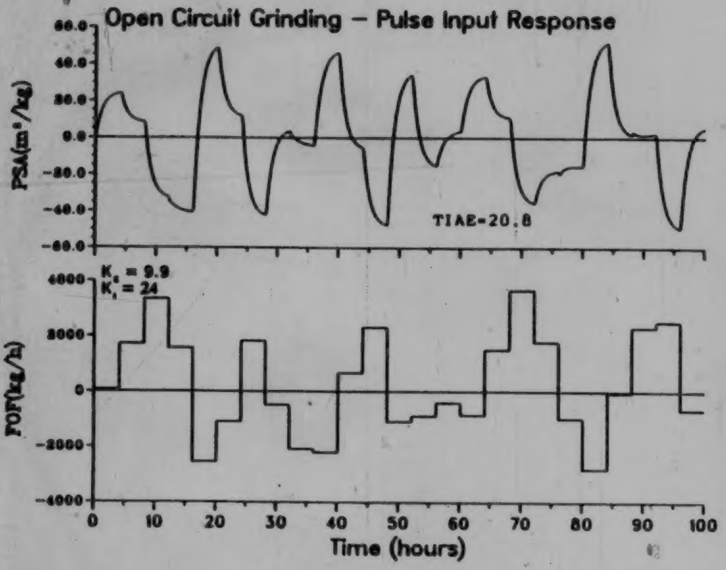


Figure 6.19: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PI Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance

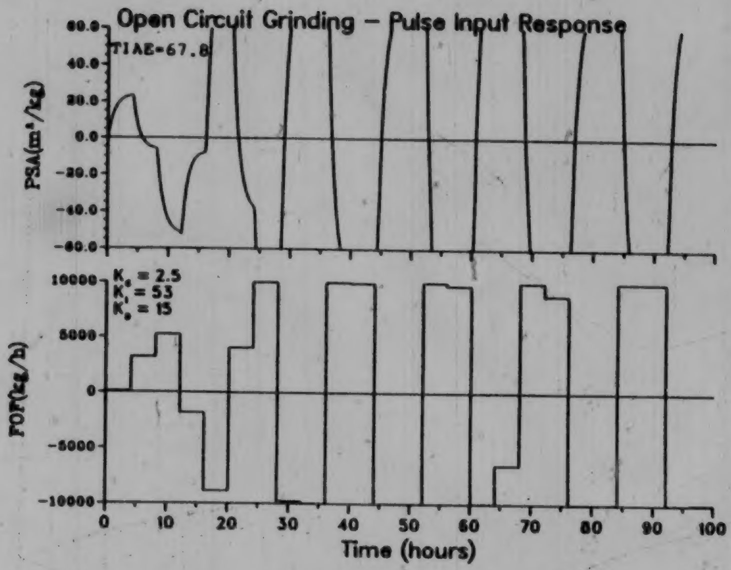


Figure 6.20: Closed Loop Response of the Open Circuit Grinding Operation using a Discrete PID Controller with a Sample Time of 4.0 Hours for a Pulse Disturbance

provides the best performance (although still quite poor), while the addition of integral action (cf. Figure 6.19) and then derivative action (cf. Figure 6.20) progressively degrades the control system performance. This behavior is probably a a result of the fact that the sample interval corresponds to the pulse duration of 4 hours. It is expected that better results could be obtained if a sample time slightly shorter than 4 hours were to be used.

7. Conclusions and Recommendations

7.1 Conclusions

The primary purpose of this work was to develop a grinding circuit simulation package and to demonstrate the usefulness of such a package to a plant engineer for tasks such as operator training and control strategy performance analysis. The software system, MINSIM, proved to be a useful tool for achieving this goal. Based on the results of this work, the following conclusions can be stated:

1. The advantages afforded by the unit operation approach to process control used in the chemical industry can be applied to the mineral processing industry. This approach is especially useful for aiding in the development of a dynamic simulator for studying control system behavior because it allows each individual piece of equipment to be modelled separately.
2. The models required for the dynamic simulation of a mineral grinding circuit, including ball and rod mills, sumps and pumps, and hydrocyclones, can be easily derived using a phenomenological modelling approach. These models are reasonably simple to understand and can provide an accurate representation of a circuit if experimental data is available to establish model parameters.
3. The DYFLO2 program structure is a convenient approach

for implementing a grinding circuit simulator because several well documented utility subroutines exist and it is based on the widely available computer language, FORTRAN IV. Each unit operation in the grinding circuit becomes a separate FORTRAN subroutine module that can be linked to other unit operations through CALLs to the appropriate subroutine. This enables many flowsheets to be simulated because any combination and number of unit operations to be linked in this manner. The MINSIM software system embodies these features.

4. To reduce computing costs, the implicit integration subroutine provided in DYFLO2 was employed in the development of the MINSIM system. Due to the highly coupled nature of the set of ordinary differential equations used to describe the behavior of grinding mills, a more advanced, iterative integration scheme for stiff systems suggested by Franks (1982) was investigated. This technique did not provide any advantage over the existing DYFLO2 stiff integration subroutine due to single precision computer computations.
5. The MINSIM software package was used to develop an interactive grinding circuit simulator for the Lake Dufault operation. This simulator made it possible to investigate the behavior of the circuit with several different multivariable control schemes, which demonstrate the usefulness of a dynamic grinding

circuit simulator for both operator training and control system analysis and design.

6. Using the simulator to model the open loop behavior of the Lake Dufault grinding circuit, it was possible to develop a 2x2 transfer function model that was representative of the nonlinear model behavior for operation near the steady state operating conditions. The transfer function parameters were established using a graphical technique suggested by Smith and Corripio (1985). The model consisted of the transfer functions that relate the controlled variables, COS and SPL, to the manipulated variables, FOF and SWF, and the disturbance variable, HRD. This model was originally developed to reduce the computation cost of the simulator, but was later used to design time delay compensation and noninteracting control strategies.
7. Conventional proportional plus integral (PI) controllers were used with the classical Type I/Type II multiloop control schemes to control hydrocyclone overflow size (COS) and sump pulp level (SPL) using fresh ore feed (FOF) rate and sump water feed (SWF) flow rate as the manipulated variables. Satisfactory control performance for both regulatory and servo control objectives was accomplished, but only after manual tuning of the controller constants (due to interaction between the control loops). It should be noted here that both the nonlinear and linear models

were used in this study, and the results showed that the linear model provided a reasonably good representation of the control system behavior at a significantly reduced simulation cost.

8. The Ogunaike and Ray multivariable time delay compensation scheme was easily implemented using the 2x2 transfer function model of the grinding circuit. This strategy, used in conjunction with the nonlinear circuit model, was not effective in rejecting ore hardness disturbances with either the Type I or Type II variable pairing scheme. Some improvement in control performance was realized when the linear circuit model was substituted for the nonlinear circuit model. This indicates that model mismatch exists between the nonlinear and linear models.
9. The Ogunnaike and Ray time delay compensation technique had a large effect on the setpoint tracking performance of the grinding circuit. Marked improvements in control were noted for both the nonlinear and linear circuit models using either the Type I or Type II variable pairing scheme.
10. Simulation results for both Type I and Type II system using the nonlinear circuit model showed that use of static decoupling was not effective in improving the regulatory or servo control performance. This is due to the time delays in the system.
11. Use of a noninteracting control strategy utilizing

dynamic decoupling controllers was found to be more effective than use of static decouplers. Due to the lead time dominance of the dynamic decoupling controllers designed from the linear model for the Type I variable paring scheme, tuning was necessary to obtain satisfactory control performance. Although the ore hardness disturbance rejection capabilities of the control system were not improved much, servo control of the hydrocyclone overflow size specification using the modified dynamic decoupling controllers resulted in a significant improvement in control performance. The dynamic decoupling controllers designed from the linear model for the Type II variable pairing scheme were used directly after ignoring the predictive terms that resulted from the time delays. As with the Type I case, only the servo control performance showed improvement.

12. The multivariable self tuning control system was shown to provide marginally adequate control for both the Type I and Type II vairable pairing schemes. The model parameters converge very rapidly during the identification phase while the system is under discrete fixed PI control in both cases. The poor performance of the self tuner when compared to the other control schemes has been attributed to the fact that the controller constants used to calculate the Q weighting polynomial were detuned to obtain satisfactory

discrete PI control performance during the identification stage.

13. The MINSIM software package was used to simulate the open circuit grinding operation described by Adel et al. (1983). Simulation of the circuit made it possible to demonstrate some additional features of the MINSIM system including the application of continuous versus discrete P, PI, and PID controllers to such a circuit.

14. The effect of analyzer time delays on control performance showed that increased analyzer time delays destabilize the grinding circuit during disturbances when under continuous feedback control. PI controllers are more sensitive to time delays than P only controllers, however, PI controllers provide the advantage of removing steady state offset from the controlled variable. PID control produces the best control system performance, using controller settings estimated using the Ziegler Nichols quarter decay ratio tuning formulae, as expected. Utilizing discrete controllers produces similar trends to those observed for the continuous controllers, however, the sample interval affects the quality of the control perfromance.

15. Simulations using controller constants estimated from the Ziegler Nichols formulae modified to account for the sample interval indicate that a proportional controller marginally outperforms a proprotional plus

integral controller for the train of four hour duration pulse disturbances. This is misleading because the four hour nominal duration of the pulse disturbance used in this study does not allow the PI controller enough time to completely remove the controlled variable offset before the next disturbance enters the system. Simply increasing the gain of the PI controller results in an improvement in performance when compared to the P only case.

16. Although the open circuit grinding simulator has been developed independently from the closed circuit simulator, it would be possible to combine the simulators by incorporating a further circuit as a second level executive option in the closed circuit simulator for convenience. This was not done in this work because it maintains the different identities of the two types of grinding circuit operations.

7.2 Recommendations

On the basis of this study, areas of activity suggested for future work include the following:

1. The capabilities of the MINSIM software package should be extended by the development of additional subroutines to model other mineral processing operations such as crushers and floatation circuits for

incorporation into the subroutine library, S.MIN.LIB.

2. The various methods for modelling changes in ore hardness should be examined in detail. The multicomponent feed stream appears to be the most promising technique to improve the accuracy of the modelling of this type of disturbance. It should be noted that this could prove to be a major undertaking.
3. Incorporation of a plot option in the MINSIM package to automatically create trend plots of the grinding circuit controlled and manipulated variable responses. This would simplfy the task of plotting the simulation results and would aid the user in assessing the control performance.
4. Implicit integration algorithms capable of providing accurate solutions using large time steps should be investigated as a means of further reducing the computing costs associated with the nonlinear grinding circuit model. Double precision computer calculations could provide the increase in accuracy required to enable a sizable increase in the integration time step. This should be carefully looked at before implementation because the entire MINSIM software system must be upgraded to be compatable with double precision computations.
5. Development of a higher order transfer function model of the Lake Dufault grinding circuit to attempt to reduce the model mismatch problems discovered in the

time delay compensation simulation study. This will require a more sophisticated time delay compensation subroutine capable of handling the higher order dynamics of the transfer function model. In addition, this higher order model could be used to design a noninteracting control system to improve the regulatory behavior of the grinding circuit.

6. Self-tuning control performance using different Q weighting polynomials should be investigated.
7. The effects of process measurement noise on the performance of both the Lake Dufault grinding circuit and the open circuit grinding simulators should be investigated. This will create a more realistic environment for control system performance evaluation.
8. Further analysis of the discrete control performance of the Lake Dufault grinding circuit should be undertaken to determine if any improvement in performance characteristics would result over the continuous control cases described in this work.
9. Other control strategies involving the use of other manipulated variables, such as mill and slurry pump speeds, should be considered for the Lake Dufault grinding circuit.
10. The universal nature of the MINSIM system should further be demonstrated by simulating grinding circuits currently under study by others such as the Duval circuit described by Bascur, Freeh and Herbst (1985)

and the East Driefontein operation discussed by Hulbert (1983). This would aid in comparing the features of the MINSIM system with other published mineral circuit simulators such as DYNAMILL (Rajamani and Herbst, 1980).

8. Nomenclature

Technical Abbreviations

COS - hydrocyclone overflow size (%-100 mesh)
FHC - friction head contribution
FOF - fresh ore feed (tonnes/hour)
FSA - feed specific area (m^2/kg)
GRD - grindability (m^2/kWh)
HRD - ore hardness
P - proportional controller
PHC - pressure head contribution
PI - proportional plus integral controller
PID - proportional, integral, derivative controller
PSA - product specific area (m^2/kg)
RDH - resistance dynamic head
RGA - relative gain array
SHC - static head contribution
SPL - sump pulp level (m)
SWF - sump water feed (tonnes/hour)
TDH - total dynamic head
TIAE - time averaged integral of the absolute error

Variables

A - attributes of variable time delay at interface
or the open loop steady state gain matrix
B - response of output variable (%)
or the breakage function
or the closed loop steady state gain matrix
C - feedback signal input to the controller comparator
D - mass flow rate of discharge stream (tonnes/h)
or the decoupling transfer function matrix
F - mass flow rate of feed stream (tonnes/h)
G - transfer function
H - mass holdup (tonnes)
or measurment device transfer function
I - attributes of variable time delay input stream
or the identity matrix
K - arbitrary constant or
controller or process gain
L - sump level measured from sump discharge (m)
or the Laplace transform of a load input
M - magnitude of input variable step change (tonnes/h or %)
or ore hardness multiplier
N - number of blocks in variable time delay (unitless)
O - attributes of variable time delay output stream
P - pressure at hydrocyclone inlet (kPa)
or mass flow rate of product stream (tonnes/h)
or the process transfer function matrix

or the self-tuning controller process output weighting polynomial
Q - volumetric flow rate of pulp (m^3/h)
or the load transfer function vector
or the self-tuning controller Q weighting polynomial
R - Laplace transform of the setpoint
or the self-tuning controller setpoint weighting polynomial
S - total solids mass holdup of the sump (tonnes)
T - time constant
U - Laplace transform of the controller output signal
V - volumetric holdup (m^3)
or measurable disturbances
W - mass flow rate of pure water (tonnes/h)
X - Laplace transform of an input signal
Y - mass fraction of classifier feed solids that report to the underflow stream or
the Laplace transform of the process output signal
Z - lift height (m)
Ff - mass fraction in feed stream
Rf - mass fraction of water from hydrocyclone feed recovered in underflow stream
b - discretized breakage function, indicate fraction of particles broken from one size class that reports to another
c - concentration
d - diameter
e - exponential operator
f - frequency of particles in feed stream
g - time domain representation of a transfer function
or gravitational constant (m/s^2)
i - size classification number
j - size classification number always larger than i
k - selection function, indicates the rate of particle breakage ($minutes^{-1}$)
or discrete time interval
p - mass frequency of particles in product stream
r - setpoint or ratio
s - Laplace operator
t - time (minutes)
u - controller output signal
v - volumetric holdup of a block in the variable time delay (m^3)
x - particle diameter (μ)
or an input signal
y - process output signal
z^{-1} - backshift operator

Greek

Δ - indicates a finite increment, measurable, but small

E - Laplace transform of the error signal
Ξ - random noise
Σ - summation
α - rate amplitude of a continuous PID controller
γ - hydrocyclone sharpness of separation modulus
δ - partial differential operator
ϵ - error signal
λ - interaction measure
τ - mean residence time of pulp in mill (minutes)
or time constant (minutes)
ρ - density (tonnes/m^3)

Superscripts

d - discrete measurable disturbance delay time
i - size class number
j - size class number
k - discrete delay time
α - breakage function distribution modulus
or delay time
' - indicates intermediate value or step
* - indicates decoupling controller or predicted signal
^ - indicates estimate

Subscripts

C - controller
D - derivative time
I - integral time
L - Laplace transform of load
a - actual
b - bias
c - corrected for bypass
d - delay time
f - feed stream
i - for size class number
j - variable time delay index number
k - indicates time delay compensator
l - load
m - maximum or measurement
n - last size class number (pan fraction)
o - hydrocyclone overflow stream
p - pulp or process
r - residual
s - solids or sample time
u - ultimate
w - water
ff - feedforward

ij - from size class j to size class i
md - measurment device
ol - open loop
sp - setpoint
50 - equiprobable partition

Other

Bold print - indicates matrix

9. REFERENCES

Adel, G.T.; "An Interactive Simulation Package For Mineral Processing Systems"; D.Eng. Thesis, University of California, Berkeley, 1982.

Adel, G.T., Ulsoy, A.G., and Sastry, K.V.S.; "A Theoretical Analysis and Control Study of Open-Circuit Grinding"; *Int. J. Min. Proc.*, Vol. 10, No. 1, 1983; pp. 25-43.

Alevisakis, G. and Seborg, D.E.; An Extension of the Smith Predictor Method to Multivariable Linear Systems Containing Time Delays", *Int. J. Control*, Vol. 3, 1973; p. 541.

Alevisakis, G. and Seborg, D.E.; "Control of Multivariable Systems Containing Time Delays Using a Multivariable Smith Predictor", *Chem. Eng. Sci.*, Vol. 29, 1974; p 373.

Allee, R.Y., Burdell, B.W., and Dison, D.L.; "CMC Concentrator Control System"; *ISA*, Paper #85-0289, 1985; pp. 171-175.

Arterburn, R.A.; "The Sizing and Selection of Hydrocyclones"; in *Design and Installation of Comminution Circuits*, A.L. Mular and G.V. Jergensen, II eds, SME of AIME, New York, 1982; pp. 592-607.

Austin, L.G., and Klimpel, R.R.; "Modeling for Scale-up of Tumbling Mills"; *Control'84*, 1984; pp. 167-184.

Barker, I.J. and Hulbert, D.G.; "Dynamic Behavior in the Control of Milling Circuits"; *IFAC Automation in Mining, Mineral and Metal Processing, Helsinki, Finland*, 1983; pp. 139-152.

Barrett-Lennard, I., and Blair, J.R.; "Multi-Variable Identification and Optimal Control for Bauxite Digestion"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 249-255.

Bascur, A.O., Freeh, E.J., and Herbst, J.A.; "Dynamic Simulation for Estimation of Industrial Grinding Circuit Control Characteristics"; *ISA*, Paper #85-0288, 1985; pp. 161-170.

Boxenbom, A.S. and Hood, R.; "General Algebraic Method Applied to Control Analysis of Complex Engine Types", Report NCA-TR-980, National Advisory Committee for Aeronautics, Washington, D.C., 1949.

Bristol, E.W.; "On a New Measure of Interaction for Multivariable Process Control"; *IEEE Trans. Auto. Control*, Vol. 11, 1966; pp.133-134.

Cohen , G.H. and Coon, G.A.;, "Theoretical Considerations of Retarded Control"; *Transactions of the ASME*, July, 1953; pp. 827-834.

Coughanowr, D.R. and Koppel, L.B.; *Process Systems Analysis and Control*, McGraw-Hill, 1965.

del Villar, R., Laplante, A.R.; "Grinding Simulation in Applesoft Basic"; *CIM Bulletin*, Vol. 78, No. 11, 1985; pp. 62-65.

Ferrara, G., Guarascio, M., and Schena, G.; "Modelling and Simulation of Integrated Plant Operations of Mineral Processing"; *Control'84*, 1984; pp. 153-165.

Fewings, J.H., Wickham, P., and Wiseman, D.M.; "Commissioning of a Digital Computer Based Wet Mineral Grinding Circuit Control System"; *Automation in Mining, Mineral and Metal Processing, Proc. 2nd IFAC Symp.* 13-17, September, 1976; pp. 105-115.

Finlayson, R.M., and Hulbert, D.G.; "The Simulation of the Behavior of Individual Minerals in a Closed Grinding Circuit"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 323-332.

Flintoff, B.C., Wong, W.Y., and Wood, R.K.; "Dynamic

Simulation of Grinding Circuit Control"; *ISA Transactions*, Vol. 24, No. 1, 1985; pp. 11-22.

Fournier, R.D. and Smith, H.W.; "Experimental Parameter Identification for a Dynamic Model of a Continuous Rod Mill"; *CIM Bulletin*, Vol. 65, No. 12, 1972; pp. 79-82.

Franks, R.G.E.; *Modeling and Simulation in Chemical Engineering*; John Wiley and Sons, Inc., New York, 1972.

Franks, R.G.E.; "DYFLO Update: DYFLO2", *Proc. Summer Conf. Comp. Simul.*, Denver, 1982; p. 507.

Fuerstenau, D.W., Venkataraman, K.S., and Williams, M.C.; "Simulation of Closed-Circuit Mill Dynamics"; *Control'84*, 1984; pp. 49-53.

Gray, G.T.; "The Developement of the Software for a Microprocessor-Based Multivariable Controller"; *Proc. of the IFAC 7th Conf. on Digital Computer Applications to Process Control*; Vienna, Austria, 1985; pp. 517-522.

Guillermo, G.R., Hernán, M.S., and Fernando de Mayo, I.; "A Dynamic Compensation for Static Particle Size Distribution Estimators"; *ISA*, Paper #85-0290, 1985; pp. 177-181.

Hammoude, A., and Smith, H.W.; "Experiments with Self-Tuning

Control of Flotation"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada,* 1980; pp. 213-218.

Herbst, J.A. and Bascur, O.A.; "Mineral-Processing Control in the 1980's -- Realities and Dreams"; *Control'84,* 1984; pp. 197-215.

Herbst, J.A. and Fuerstenau, D.W.; "Scale-Up Procedure For Continuous Grinding Mill Design Using Population Balance Models"; *International Journal of Mineral Processing,* Vol. 7, 1980; pp. 1-31.

Herbst, J.A., Rajamani, K., and Pate, W.T.; "Identification of Ore Hardness Disturbances in a Grinding Circuit Using a Kalman Filter"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada,* 1980; pp. 333-348.

Herbst, J.A., Robertson, K., and Rajamani, K.; "Mill Speed as a Manipulated Variable for Ball Mill Grinding Control"; *IFAC Automation in Mining, Mineral and Metal Processing, Helsinki, Finland,* 1983; pp. 153-160.

Hinde, A.L., Lloyd, P.J.D., Mackay, J.G. and King, R.P.; "The Use of Stochastic Modelling Techniques to Describe the Dynamics of a Gold Milling Circuit"; *Automation in Mining, Mineral and Metal Processing, Proc. 2nd IFAC Symp.* September 13-17, 1976; pp. 565-575.

Hulbert, D.G.; "Multivariable Control and Optimization of the Operation of a Milling Circuit at East Driefontein Gold Mine"; MINTEK Report No. M98, May, 1983.

Hulbert, D.G., Braae, M.; "Multivariable Control of a Milling Circuit at East Driefontein Gold Mine"; National Institute for Metallurgy, Report No. 2113, August, 1981.

Hulbert, D.G., Koudstaal, J., Braae, M., and Gossman, G.I.; "Multivariable Control of an Industrial Grinding Circuit"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 311-322.

Hulbert, D.G. and Woodburn, E.T.; "Multivariable Control af a Wet-Grinding Circuit"; *AIChE Journal*, Vol. 29, No. 2; pp. 186-191.

Isermann, R.; *Digital Control Systems*; Springer-Verlag, New York, 1981.

Jamsa, S.-L., Melama, H., and Penttinen, J.; "Design and Experimental Evaluation of a Multivariable Grinding Circuit Control System"; *IFAC Automation in Mining, Mineral and Processing, Helsinki, Finland*, 1983; pp. 189-197.

Langman, J.A.;"Q Weighting and Self-Tuning Control of a Binary Distillation Column", M.Sc. Thesis, Department of

Chemical Engineering, University of Alberta, Edmonton, 1987.

Lean, P.J. and Baker, D.C.; "Grinding Circuit Control at the New Broken Hill Consolidated Concentrator, Broken Hill, Australia"; *Control'84*, 1984; pp. 243-251.

Lee, J.E.; "Slurry Pumping at Inco Metals Company (Ontario Division)"; *Can. Min.*, Vol. 99, No. 3, 1978; p. 14.

Lee, P.L. and Newell, R.B.; "Multivariable Control of a Grinding Circuit"; *Proc. AMRD - IFAC, Amsterdam*, 1985; pp. 283-287.

Luyben, W.L.; "Distillation Decoupling"; *AIChE Journal*, Vol. 16, No. 2, March, 1970; pp. 198-203.

Lynch, A.J.; "The Automatic Control of Mineral Preparation and Concentration Circuits"; *Control'84*, 1984; pp. 3-12.

Lynch, A.J.; *Mineral Crushing and Grinding Circuits: Their Simulation, Optimisation, Design and Control*, Elsevier Scientific Publishing Company, 1977.

Marchand, J.C., Hodouin, D., and Everell, M.D.; "Residence Time Distribution and Mass Transport Characteristics of Large Industrial Grinding Mills"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 295-302.

McElvain, R.E. and Cave, I.; "Selection and Sizing of Slurry Pumps for Grinding Circuits"; in *Design and Installation of Comminution Circuits*, A.L. Mular and G.V. Jergensen II, eds; SME of AIME, New York, 1982; pp. 573-591.

Ogunnaike, B.A. and Ray, W.H.; "Multivariable Controller Design for Linear Systems Having Multiple Time Delays"; *AIChE Journal*, Vol. 25, No. 6, 1979; pp. 1043-1057.

Phillips, C.L., and Nagle, H.T. Jr.; *Digital Control System Analysis and Design*; Prentice Hall Inc., Englewood Cliffs, N.J., 1984.

Plitt, L.R.; "A Mathematical Model of The Hydrocyclone Classifier"; *CIM Bulletin*, Vol. 69, No. 12, 1976; pp. 114-123.

Ragot, J., Roesch, M., Degoul, and Berube, Y.; "Transient Study of a Closed Grinding Circuit"; *Automation in Mining, Mineral and Metal Processing, Proc. 2nd IFAC Symp.* 13-17, September, 1976; pp. 129-142.

Rainville, E.D.; Bedient, E.P.; *Elementary Differential Equations*; 5th Ed., Macmillan Publishing Co., Inc., New York, 1974.

Rajamani, K.; "Self-Tuning Control of a Ball Mill Grinding

Circuit"; *ISA*, Paper #85-0287, 1985; pp. 153-159.

Rajamani, K. and Herbst, J.A.; "A Dynamic Simulator for the Evaluation of Grinding Circuit Control Strategies"; *Proc. Eur. Symp. on Part. Technology, Amsterdam*, 1980; pp. 64-81.

Romberg, T.M., and Jacobs, W.S.V.; "Identification of a Mineral Processing Plant from Normal Operating Data"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 275-282.

Rowland, C.A. Jr.; "Selection of Rod Mills, Ball Mills, Pebble Mills and Regrind Mills"; in *Design and Installation of Comminution Circuits*, A.L. Mular and G.V. Jergensen II, eds; SME of AIME, New York, 1982; pp. 393-438.

Sastry, K.V.S. and Adel, G.T.; "A Survey of Computer Simulation Software for Mineral Processing Systems"; *Control'84*, 1984; pp. 121-130.

Seitz, R.A. and Kawatra, S.K.; "Further Studies on the Use of Classifiers for the Control of Wet Grinding Circuits"; *Int. Journ. of Min. Proc.*, Vol. 12, 1984; pp. 239-249.

Smith, C.A., Corripio, A.B.; *Principles and Practice of Automatic Process Control*, John Wiley and Sons, Inc., New York, 1985.

Smith, H.W. and Guerin, D.; "Dynamic Modelling, Simulation and Control of a Grinding Circuit"; *CIM Bulletin*, Vol. 73, No. 9, 1980; pp. 133-138.

Smith, H.W. and Lewis, C.L.; "Computer Control Experiments at Lake Dufault"; *CIM Bulletin*, Vol. 62, No. 2, 1969; pp. 109-115.

Stephanopoulos, G.; *Chemical Process Control - An Introduction to Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1984.

Su, Z., Yan, W.; "Optimal Search and Self-Adaptive Control for Flotation"; *Control'84*, 1984; pp. 235-238.

Tung, L.S. and Edgar, T.F.; "Analysis of Control - Output Interactions in Dynamic Systems"; *AIChE Journal*, Vol. 27, No. 4, 1981; pp. 690-693.

Vogel, E.F.; Edgar, T.F.; "An Adaptive Dead Time Compensator for Process Control"; *Proc. 1980 Joint Automatic Control Conference*, August 13-15, San Francisco.

Wade, H.L. and Chang, J.W.; "A Decoupling Technique With Application In Grinding Control"; *Instrumentation in the Mining and Metallurgy Industries*, Vol. 4, 1976; pp. 1-18.

Weller, K.R.; "Hold-up and Residence Time Characteristics of Full Scale Grinding Circuits"; *IFAC Mining, Mineral and Metal Processing, Montreal, Canada*, 1980; pp. 303-309.

Witcher, M.F. and McAvoy, T.J.; "Interacting Control Systems: Steady State and Dynamic Measurement of Interaction"; *ISA Transactions*, Vol. 16, No. 3, 1977; pp. 35-41.

Wong, W.Y.; M.Eng. Project Report, Department of Chemical Engineering, University of Alberta, 1984.

Wood, R.K.; "Improved Control By Application of Advanced Control Techniques"; *ISA Transactions*, Vol. 16, No. 4, 1977; pp. 31-39.

Wood, R.K.; "Modeling and Control of Crushing and Grinding Circuits"; Presented at the 30th Annual ISA Conf., October 6-9, 1975, Milwaukee, Wisconsin.

Wood, R.K., and Flintoff, B.C.; "Dynamic Process Simulation: CSSL vs. Process Simulation Language", *Proc. Summer Conf. Simul. Denver*, 1982, p. 492.

Wyatt-Mair, G.F., Garner, K.C., and King, R.P.; "Realtime Digital Computer State Estimator for a Hard Rock Milling Circuit"; *IFAC Mining, Mineral and Metal Processing,* *Montreal, Canada*, 1980; pp. 363-374.

Zalkind, C.S.; "Practical Approach to Non-Interacting Control Part I"; *Instruments and Control Systems*, Vol. 40, No. 3, 1967; pp. 89-93.

Zalkind, C.S.; "Practical Approach to Non-Interacting Control Part II"; *Instruments and Control Systems*, Vol. 40, No. 4, 1967; pp. 111-116.

Ziegler, J.G., and Nichols, N.B.; "Optimum Settings for Automatic Controllers"; *Transactions of the ASME*, November, 1942; pp. 759-768.

Appendix A: Subroutine Library Descriptions

A.1 DYFLO2.LIB (Extended)

A.1.1 PROGRAM BDATA

Purpose: To initialize the DYFLO2 COMMON BLOCKs.

NOTE:The BLOCK DATA program must be compiled and used in the run command.

A.1.2 SUBROUTINE CONTR1

Purpose: Proportional (P) only controller subroutine.

Use: `CALL CONTR1 (VI,CO,ZR,RNG,SP,AXN,PB,OI)`

Parameters:

VI - INPUT SIGNAL
CO - CONTROLLER OUTPUT
ZR - ZERO READING
FSRDG - FULL SCALE READING
SP - SETPOINT
AXN - ACTION (DIRECT +1, REVERSE -1)
PB - PROPORTIONAL BAND
OI - BIAS

NOTE: The bias (OI) must be initialized before the first call to this routine. This can be calculated as follows:
OI = STEADY STATE VALUE OF OUTPUT IN % OF FULL SCALE

A.1.3 SUBROUTINE CONTR2

Purpose: Proportional plus integral (PI) controller subroutine.

Use: `CALL CONTR2(VI,CO,ZR,RNG,SP,AXN,PB,RPT,OI)`

Parameters:

VI - INPUT SIGNAL
CO - CONTROLLER OUTPUT
ZR - ZERO READING
FSRDG - FULL SCALE READING
SP - SETPOINT
AXN - ACTION (DIRECT +1, REVERSE -1)
PB - PROPORTIONAL BAND
OI - INTEGRAL ACTION
RPT - RESET RATE, REPEATS PER TIME

NOTE: The integral action (OI) must be initialized before

the first call to this routine. This can be calculated as follows:
OI = STEADY STATE OUTPUT VALUE IN % OF FULL SCALE

A.1.4 SUBROUTINE CONTR3

Purpose: Proportional plus integral plus derivative (PID) controller subroutine.

Use: CALL CONTR3(VI,CO,ZR,RNG,SP,AXN,PB,RPT,RT,RA,OI,ODI)

Parameters:

VI - INPUT SIGNAL
CO - CONTROLLER OUTPUT
ZR - ZERO READING
FSRDG - FULL SCALE READING
SP - SETPOINT
AXN - ACTION (DIRECT +1, REVERSE -1)
PB - PROPORTIONAL BAND
RPT - RESET RATE, REPEATS PER TIME
RT - RATE TIME
RA - RATE AMPLITUDE
OI - INTEGRAL ACTION
ODI - OUTPUT OF DERIVATIVE ACTION

NOTE: Both the integral action (OI) and the derivative action (ODI) must be initialized before the first call to this routine. These two variables can be calculated as follows:
OI = STEADY STATE OUTPUT VALUE IN % OF FULL SCALE
ODI = 1./RA-1.

A.1.5 SUBROUTINE CON2X2

Purpose: Continuous 2x2 controller routine

Use: CALL CON2X2(CO1,CO2,CI1,CI2,MODE1,MODE2)

Parameters:

CI1 - LOOP 1 CONTROLLER INPUT
CI2 - LOOP 2 CONTROLLER INPUT
MODE1 - LOOP 1 CONTROLLER MODE 1---> P
2---> PI
3---> PID
MODE2 - LOOP 2 CONTROLLER MODE 1---> P
2---> PI
3---> PID
CO1 - LOOP 1 CONTROLLER OUTPUT (0 - 100%)
CO2 - LOOP 2 CONTROLLER OUTPUT (0 - 100%)

A.1.6 SUBROUTINE DER

Purpose: Numerical derivative subroutine.

Use: CALL DER(Y,X,DYDX,I)

Parameters:

Y - VARIABLE Y
X - VARIABLE X
DYDX - DERIVATIVE OF Y WITH RESPECT TO X
I - ROUTINE CALL INDEX

A.1.7 SUBROUTINE DISPID

Purpose: Discrete PID controller using the velocity formulation with the derivative action only on the input signal subroutine.

Use: CALL DISPID(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,
TIM,N,TO)

Parameters:

VI - process input signal
CO - controller output signal, incremental
ZR - process input zero reading
FSRDG - process input full scale reading
SP - setpoint, incremental
AXN - comparator action +1---> direct action
-1---> reverse action
KP - proportional gain
KI - integral gain
KD - derivative gain
TS - sample time
MODE - controller mode 1---> P
2---> P+I
3---> P+I+D
TIM - simulation time
N - discrete PID controller number
TO - sample time offset

A.1.8 SUBROUTINE DPID2

Purpose: Discrete PID Controller using the velocity formulation with the standard textbook structure subroutine.

Use: CALL DPID2(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,
[illegible]IM,N)

Parameters:

VI - process input signal
CO - controller output signal, incremental
ZR - process input zero reading
FSRDG - process input full scale reading
SP - setpoint, incremental
AXN - comparator action +1---> direct action
-1---> reverse action
KP - proportional gain
KI - integral gain
KD - derivative gain
TS - sample time
MODE - controller mode 1---> P
2---> P+I
3---> P+I+D
TIM - simulation time
N - discrete PID controller number

A.1.9 SUBROUTINE FFCALC

Purpose: Forcing function calculation subrpotine.

Use: CALL FFCALC(X,FFVECT)

NOTE: This is a problem dependent subroutine. The forcing function equations must be coded by the user.
Parameters:

X - INDEPENDENT VARIABLE VECTOR
FFVECT - FORCING FUNCTION VECTOR

A.1.10 SUBROUTINE GJR

Purpose: Solution of simultaneous algebraic equations using Gause-Jordan reduction.

Use: CALL GJR(A,N,TOL,KEY,X,DETER)

Parameters:

A - IS AUGMENTED MATRIX (N,N+1) ** EXAMPLE FOR N=2
A(1,1)*X1+A(1,2)*X2=A(1,3)
A(2,1)*X1+A(2,2)*X2=A(2,3)
TOL - SOLUTION TOLERANCE (=.1E-6)
KEY - SOLUTION OBTAINED FLAG (=1 FOR SOLUTION)
X - SOLUTION VECTOR
DETER - OUTPUT DETERMINANT

A.1.11 SUBROUTINE INTCPD

Purpose: Coupled system ordinary differential equation integration subroutine for up to 30 ODE's.

Use: CALL INTCPD(N,CMAX,CMIN,X,W)

Parameters:

N - NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM
CMAX - MAXIMUM CHANGE IN INDEPENDENT VARIABLE AS A FRACTION OF THE INDEPENDENT VARIABLE ESTIMATE
CMIN - MINIMUM CHANGE IN INDEPENDENT VARIABLE AS A FRACTION OF THE INDEPENDENT VARIABLE ESTIMATE
X - VECTOR CONTAINING INDEPENDENT VARIABLES
W - RELAXATION FACTOR VECTOR

A.1.12 SUBROUTINE INTG

Purpose: Dependent variable general integration subroutine. This routine is capable of performing first order Euler integration, and second and fourth order Runge-Kutta integration.

Use: CALL INTG(X,DX)

Parameters:

X - DEPENDENT VARIABLE
DX - DERIVATIVE OF X

A.1.13 SUBROUTINE INTI

Purpose: Integration control subroutine. This routine integrates the independent variable.

Use: CALL INTI(TD,DTD,IOD)

Parameters:

TD-INDEPENDENT VARIABLE NAME
DTD-INTEGRATION STEP SIZE
IOD-INTEGRATION METHOD
  IOD=1 FIRST ORDER EULER
  IOD=2 SECOND ORDER RUNGE KUTTA
  IOD=3 FOURTH ORDER RUNGE KUTTA

A.1.14 SUBROUTINE INTSTF

Purpose: Implicit integration subroutine for equation in the form of DX/DT=FF-DK*X

Use: CALL INTSTF(X,TK,FF)

Parameters:

X - DEPENDENT VARIABLE
TK - PECIPROCAL TIME CONSTANT
FF - FORCING FUNCTION

A.1.15 SUBROUTINE INTST2

Purpose: Implicit integration subroutine for equation in the form of DX/DT=FF-DK*X**2

Use: CALL INTST2(X,DK,FF)

Parameters:

X - DEPENDENT VARIABLE
DK - RECIPROCAL TIME CONSTANT
FF - FORCING FUNCTION

A.1.16 SUBROUTINE I2BY2

Purpose: An interacting 2x2 first order transfer function model routine

Use: CALL I2BY2(TIM,RINT,N)

Parameters:

TIM - CURRENT SIMULATION TIME
RINT - INTEGRATION INTERVAL
N - ROUTINE CALL NUMBER

A.1.17 SUBROUTINE JFUNC

Purpose: Function evaluation subroutine for routine NUMJAC.

Use: CALL JFUNC(X,XOLD,N,F)

Parameters:

X - INDEPENDENT VARIABLE VECTOR
XOLD - PREVIOUS INDEPENDENT VARIABLE VECTOR
N - NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM
F - FUNCTION VALUE VECTOR

A.1.18 SUBROUTINE LLCDER

Purpose: Lead-lag compensator subroutine using the numerical derivative implementation

Use: CALL LLCDER(YOUT,XIN,TC1,TC2,TD,GAIN,TIM,NO,TNO,RINT)

Parameters:

YOUT - OUTPUT SIGNAL
XIN - INPUT SIGNAL
TC1 - LEAD TIME CONSTANT
TC2 - LAG TIME CONSTANT
TD - DEAD TIME
GAIN - COMPENSATOR GAIN
TIM - SIMULATION TIME
NO - LEAD-LAG NUMBER - MAX. 10
TNO - TIME DELAY NUMBER
RINT - INTEGRATION INTERVAL

A.1.19 SUBROUTINE LLCOMP

Purpose: Lead-lag compensator subroutine avoiding the use of the numerical derivative.

Reference: Smith & Corripio; Principles and Practice of Automatic Process Control, JOHN WILEY & SONS, 1985, pp. 497-498.

Use: CALL LLCOMP(YOUT,XIN,TC1,TC2,TD,GAIN,NO)

Parameters:

YOUT - OUTPUT SIGNAL
XIN - INPUT SIGNAL
TC1 - LEAD TIME CONSTANT
TC2 - LAG TIME CONSTANT
TD - TIME DELAY
GAIN - COMPENSATOR GAIN
NO - LEAD-LAG NUMBER (MAX. OF 25)

A.1.20 SUBROUTINE NUMJAC

Purpose: Numerical Jacobian matrix calculation subroutine.

Use: CALL NUMJAC(N,X,XOLD,F,JAC)

Parameters:

N - NUMBER OF EQUATIONS IN SYSTEM
X - INDEPENDENT VARIABLE VECTOR
XOLD - PREVIOUS TIME STEP INDEPENDENT VARIABLE VECTOR
F - FUNCTION VALUE VECTOR
JAC - JACOBIAN MATRIX

A.1.21 SUBROUTINE PRNTF

Purpose: Print simulation at a specified frequency and test for the end of the simulation run.

Use: CALL PRNTF(PRI,FNR,NF,A,B,C,D,E,F,G,O,P,Q)

Parameters:

PRI-PRINT INTERVAL
FNR-TERMINATION TIME
NF-FINISH INDEX (NF=.TRUE.,FINISHED)
A...G,O,P,Q-VARIABLES

A.1.22 SUBROUTINE PTRAIN

Purpose: Square wave pulse train generation subroutine.

Use: CALL PTRAIN(OUT,AMPL,P,PTIM,TIM,NO)

Parameters:

OUT	- output signal
AMPL	- amplitude of pulse wave
P	- normalized pulse definition vector
PTIM	- pulse period
TIM	- current simulation time
NO	- routine call index

A.1.23 SUBROUTINE RATCON

Purpose: Ratio controller subroutine.

Use: CALL RATCON(COR,CIR,RATIO)

Parameters:

CIR	- INPUT TO RATIO CONTROLLER
RATIO	- RATIO INPUT TO OUTPUT (CIR/COR)
COR	- OUTPUT FROM RATIO CONTROLLER

A.1.24 SUBROUTINE RELAX

Purpose: Smart relaxation factor calculation subroutine

Use: CALL RELAX(N,X,DX,CMAX,CMIN,W)

Parameters:

N	- NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM

X   - INDEBENDENT VARIABLE VECTOR
DX   - CHANGE IN INDEPENDENT VARIABLE VECTOR FROM N-R ITERATION
CMAX - MAXIMUM CHANGE IN INDEPENDENT VARIABLE
CMIN - MINIMUM CHANGE IN INDEPENDENT VARIABLE
W   - RELAXATION FACTOR

A.1.25 SUBROUTINE TDC

Purpose: Multivariable time delay compensation subroutine.

Reference: Ogunnaike and Ray, AIChE J. VOL 25, NO. 6
NOVEMBER 1979, P. 1043

Use: CALL TDC(OUT,IN,KPM,TAUM,NTLM,M,N)

Parameters:

OUT - OUTPUT SIGNAL VECTOR
IN   - INPUT SIGNAL VECTOR
KPM - PROCESS MODEL STEADY STATE GAIN MATRIX
TAUM - PROCESS MODEL TIME CONSTANT MATRIX
NTLM - PROCESS MODEL NUMBER OF TIME LOCATIONS MATRIX
M   - DIMENSION OF ABOVE VARIABLES EXACTLY AS IN CALLING ROUTINE
N   - ORDER OF PROCESS MODEL, MAX. 25

A.1.26 SUBROUTINE TDL

Purpose: Fixed time delay subroutine.

Use: CALL TDL(YA,NTLC,Y,YL,NTL,JC)

Parameters:

YA   -WORK AREA ARRAY
NTLC -TIME DELAY LOCATION COUNTER
Y   -INPUT SIGNAL
YL   -OUTPUT DELAY SIGNAL
NTL -NUMBER OF TIME LOCATIONS IN TIME DELAY (RATIO OF TIME DELAY TO INTEGRATION STEP SIZE)
JC   -SUBROUTINE CALL NUMBER

NOTE: NTLC must be initialized to 25*0 before this subroutine is called by the declaration:

```
INTEGER NTLC(25)/25*0/
```

A.1.27 SUBROUTINE TFN1

Purpose: First order transfer function subroutine

Use: CALL TFN1(OUT,XIN,TC,GAIN)

Parameters:

XIN - INPUT SIGNAL
OUT - OUTPUT SIGNAL
TC - TIME CONSTANT
GAIN - GAIN

A.1.28 SUBROUTINE TFN2

Purpose: Second order transfer function subroutine for the form:
XOUT/XIN=GAIN/(OMEGA**2*(S**2)+2*ZETA*OMEGA*S+1)

Use: CALL TFN2(OUT,XIN,A1,A2,GAIN)

Parameters:

OUT - OUTPUT SIGNAL
XIN - INPUT SIGNAL
A1 - OMEGA**2
A2 - 2*ZETA*OMEGA
GAIN - GAIN

A.1.29 SUBROUTINE TKCALC

Purpose: Reciprocal time constant calculation subroutine.

Use: CALL TKCALC(X,TKVECT)

NOTE: This is a problem dependent subroutine. The reciprocal time constant equations must be coded by the user.

Parameters:

X - INDEPENDENT VARIABLE VECTOR
TKVECT - INVERSE TIME CONSTANT VECTOR

A.1.30 SUBROUTINE ZOH

Purpose: Zero order hold subroutine

Use: CALL ZOH(YOUT,XIN,SAMT,TIM,NO)

Parameters:

YOUT - OUTPUT SIGNAL
XIN - INPUT SIGNAL

SAMT - SAMPLE TIME
TIM - SIMULATION TIME
NO - ZOH NUMBER - MAX. 10

A.2 MIN.LIB

A.2.1 SUBROUTINE BLEND

Purpose: This subroutine will blend up to three input streams.

Use: CALL BLEND(IN1,IN2,IN3,OUT)

Parameters:

IN1 = BLENDER INPUT STREAM NUMBER 1
IN2 = BLENDER INPUT STREAM NUMBER 2
IN3 = BLENDER INPUT STREAM NUMBER 3
(NOTE: STREAM IS IGNORED IF THE PARAMETER VALUE IS ZERO)
OUT = BLENDER OUTPUT STREAM NUMBER

A.2.2 SUBROUTINE CLASS

Purpose: This subroutine performs the calculations for a hydrocuclone classifier using the Lynch-Rao model.

Use: CALL CLASS(IN,OF,UF,NO)

Parameters:

IN = CYCLONE INPUT STREAM NUMBER
OF = CYCLONE OVERFLOW STREAM NUMBER
UF = CYCLONE UNDERFLOW STREAM NUMBER
NO = CYCLONE EQUIPMENT NUMBER

A.2.3 SUBROUTINE CONVEY

Purpose: This subroutine simulates a fixed time delay in a processing system.

Use: CALL CONVEY(IN,OUT,NO,DELAY,DELTAT)

Parameters:

IN=INPUT FLOW STREAM NUMBER
OUT=OUTPUT FLOW STREAM NUMBER
NO=PLUG FLOW ELEMENT NUMBER
DELAY=TIME DELAY
DELTAT=INTEGRATION INTERVAL

A.2.4 SUBROUTINE MILL

Purpose: This subroutine performs the calculations to set up and solve the system of differential equations used in modelling the grinding mills

Use: CALL MILL(IN,OUT,NO,PCFLAG)

Parameters:

IN = MILL INPUT STREAM NUMBER
OUT= MILL OUTPUT STREAM NUMBER
NO = MILL EQUIPMENT NUMBER
PCFLAG= PREDICTOR CORRECTOR INTEGRATION FLAG
(PCFLAG=.TRUE. IS ON)

A.2.5 SUBROUTINE MILOUT

Purpose: This subroutine prints out mill operating parameters

Use: CALL MILOUT(NO)

Parameters:

NO = THE EQUIPMENT NUMBER OF THE GRINDING MILL

A.2.6 SUBROUTINE PIPE

Purpose: This subroutine simulates a variabel time delay in a processing system.

Use: CALL PIPE(IN,OUT,NO,HLDP,IPMAX)

Parameters:

IN=INPUT FLOW STREAM NUMBER
OUT=OUTPUT FLOW STREAM NUMBER
NO=PLUG FLOW ELEMENT NUMBER
HLDP=HOLDUP VOLUME
IPMAX=NUMBER OF BLOCKS FOR THE DELAY

A.2.7 SUBROUTINE SMPPMP

Purpose: This subroutine simulates a sump and pump in an industrial grinding circuit. It assumes that the dynamic head vs flow can be approximated by a quadratic. Furthermore, for this simulation, thepump discharge line is taken to be 0.2 m (8") I.D. Other explicit assumptions on piping geometry have not yet been considered.

Use: CALL SMPPMP(IN,OUT,HNO,NO,CNO,DELTAT,PCFLAG)

Parameters:

IN=SUMP INPUT STREAM NUMBER
OUT=PUMP OUTPUT STREAM
HNO=SUMP HOLDUP NO.
NO=HOLDUP EQUIPMENT NUMBER
CNO=CYCLONE EQUIPMENT NO. FOR UNIT ATTACHED TO PUMP DISCHARGE
DELTAT=THE INTEGRATION INTERVAL
PCFLAG=PREDICTOR CORRECTOR INTEGRATION ON/OFF FLAG (PCFLAG=.TRUE. IS ON)

A.2.8 SUBROUTINE STROUT

Purpose: This subroutine prints the STREAM data structure to the simulation output file.

Use: CALL STROUT(TIME)

Parameters:

TIME= SIMULATION (OR REAL) TIME

A.2.9 SUBROUTINE VOLCNT

Purpose: This subroutine adjusts the pump box water flow rate to maintain constant volumetric flow in the cyclone feed stream (after Smith)

Use: CALL VOLCNT(IN1,IN2,CNT,SETPT)

Parameters:

IN1=SUMP INPUT STREAM #1
IN2=SUMP INPUT STREAM #2
CNT=CONTROL STREAM (ADJUSTED TO MAINTAIN VOLUME)
SETPT=VOLUMETRIC FLOW RATE SETPOINT TO CYCLONE FEED SUMP

A.3 STC.LIB

A.3.1 SUBROUTINE CONTRL

Purpose: Self tuning controller subroutine. This routine performs the calculations for a FULL BLOWN SELF TUNING CONTROLLER (STC) and is based on that used in the Dynamic Column Simulation Package (DCSP).

Reference: Langman, M.Sc. Thesis, Department of Chemical Engineering, University of Alberta, 1986.

Use: CALL CONTRL(Y,U,SP,PB,TI,DER,TS)

Parameters:

Y - CONTROLLER INPUT VECTOR
U - CONTROLLER OUTPUT VECTOR
SP - SETPOINT VECTOR
PB - PROPORTIONAL BAND VECTOR
TI - INTEGRAL TIME VECTOR
DER - DERIVATIVE TIME VECTOR
TS - SAMPLE INTERVAL VECTOR

Outside Support Requirements

MIMOIN - initialization routine
MOIST1 - STC calculation routine
MOIST2 - STC calculation and output routine
ULIMIT - controller output limiting routine
IDENT1 - RLS parameter ID routine
IDENT2 - RSR parameter ID routine
IDENT3 - RUD parameter ID routine
IDENT4 - RLM parameter ID routine
PWRITE - covariance output routine

NOTE: The DYFLO users of this routine should be aware that the convention on the controller action switch (variable ACT) is opposite that used in the DYFLO routines. The routine, CNTR4I, will convert the DYFLO convention so that it is compatable with this routine.

A.3.2 SUBROUTINE MIMOIN

Purpose: MIMOIN sets up the self-tuner. In order to do this it reads information from a variety of sources. First it initializes most things to 0 except for CA (=1) and the counter IPIS, which is set to 1. From the terminal it reads:

- do you want adaptive control on the top loop?
- when do you want adaptive control to start?
- when do you want identification to start?

MIMOIN then reads, from the data file attached to logical unit 2:

N - THE NUMBER OF EACH TYPE OF PARAMETER USED BY THE CONTROL LAW (ie G, F, H AND L)
PAR - RECEIVES THE INITIAL PARAMETER VALUES
KDEL - ASSUMED MODEL DELAYS - NOTE THAT THE ACTUAL DELAY ON THE INTERACTING CONTROL IS
KDEL(*,1)+KDEL(*,2)
AND THE DISTURBANCE DELAY IS
KDEL(*,1)+KDEL(*,3)
ITYID - TYPE OF IDENTIF. (SEE THE 4 IDENT

SUBROUTINES)
ROW - "FORGETTING FACTORS"
PINT - INITIAL VALUE OF THE COVARIANCE MATRIX (& INITIALIZE THE "MATRIX")
ITYCNT - TYPE OF CONTROLLER (1=STC)
UBL - LOWER CONTROL LIMIT
UTL - UPPER CONTROL LIMIT
IRAMP - MAX. % CHANGE IN CONTROL PER STEP
ACT - CONTROL ACTION (+VE OR -VE)
NP, NQ, NR - # OF P'S, # OF Q'S AND # OF R'S

Use: CALL MIMOIN(TS,IMODE,ICONT,ICON)

Parameters:

TS - SAMPLE INTERVAL VECTOR
IMODE - CONTROLLER MODE (1-P, 2-PI, 3-PID)
ICONT - CONTROL LOOP CONFIGURATION (1-LOOP 1 CLOSED, 2-LOOP 2 CLOSED, 3-BOTH CLOSED)
ICON - CONTROL SYSTEM CONFIGURATION FLAG FROM GRINDING CIRCUIT SIMULATOR

A.3.3 SUBROUTINE MOIST1

Purpose: Self tuning controller calculation subroutine.

Use: CALL MOIST1(IS)

Parameters:

IS - LOOP NUMBER

A.3.4 SUBROUTINE MOIST2

Purpose: Self tuning controller calculation subroutine.

Use: CALL MOIST2(IS)

Parameters:

IS - LOOP NUMBER

A.3.5 SUBROUTINE ULIMIT

Purpose: This subroutine limits the control action per control interval to a specified maximum.

Use: CALL ULIMIT(UN, UNT, UTL, UBL, IRAMP)

Parameters:

UN - ADJUSTED (LIMITED) CONTROLLER OUTPUT
UNT - UNBOUNDED CONTROLLER OUTPUT
UTL - MAXIMUM CONTROLLER OUTPUT
UBL - MINIMUM CONTROLLER OUTPUT
IRAMP - ABSOLUTE VALUE OF THE MAXIMUM CHANGE IN CONTROL ACTION PER CONTROL INTERVAL

A.3.6 SUBROUTINE IDENT1

Purpose: Recursive Least Squares (RLS) parameter estimation subroutine.

Use: CALL IDENT1(YNEW, N, IS, PERER)

Parameters:

YNEW - NEW PARAMETER ESTIMATES
N - NUMBER OF PARAMETER TO ESTIMATE
IS - LOOP NUMBER
PERER - PREDICTION ERROR

A.3.7 SUBROUTINE IDENT2

Purpose: Recursive Least Squares with Square Root parameter identification subroutine

Use: CALL IDENT2(YNEW, N, IS, PERER)

Parameters:

YNEW - NEW PARAMETER ESTIMATES
N - NUMBER OF PARAMETER TO ESTIMATE
IS - LOOP NUMBER
PERER - PREDICTION ERROR

A.3.8 SUBROUTINE IDENT3

Purpose: Recursive Least Squares with Upper Diagonal factorization parameter identification subroutine.

Use: CALL IDENT3(YNEW, N, IS, PERER)

Parameters:

YNEW - NEW PARAMETER ESTIMATES
N - NUMBER OF PARAMETER TO ESTIMATE
IS - LOOP NUMBER

PERER - PREDICTION ERROR

A.3.9 SUBROUTINE IDENT4

Purpose: Recursive Learning parameter identification subroutine.

Use: CALL IDENT4(YNEW, N, IS, PERER)

Parameters:

YNEW - NEW PARAMETER ESTIMATES
N - NUMBER OF PARAMETER TO ESTIMATE
IS - LOOP NUMBER
PERER - PREDICTION ERROR

A.3.10 SUBROUTINE PWRITE:

Purpose: This subroutine prints the covariance matrix to an output file.

Use: CALL PWRITE(ITYPE, N, IS, IWIO, IPRINT)

Parameters:

ITYPE - IDENTIFICATION TYPE
N - NUMBER OF PARAMETERS
IS - LOOP NUMBER
IWIO - LOGICAL UNIT NUMBER ASSOCIATED WITH OUTPUT FILE
IPRINT - PRINT FLAG

A.3.11 SUBROUTINE CNTR4I

Purpose: Self tuning controller initialization subroutine.

Use: CALL CNTR4I(RINT,IDT,CO,ZR,FSRDG,SP,AXN,PB,TI,DER,TS)

Parameters:

RINT - INTEGRATION INTERVAL
IDT - INTEGER NUMBER OF INTEGRATION INTERVALS PER CONTROL INTERVAL
CO - CONTROLLER OUTPUT VECTOR
ZR - CONTROLLER ZERO SCALE READING VECTOR
FSRDG - CONTROLLER FULL SCALE READING VECTOR
SP - SETPOINT VECTOR
AXN - CONTROLLER ACTION VECTOR
PB - PROGORTIONAL BAND VECTOR
TI - INTEGRAL TIME VECTOR
DER - DERIVATIVE TIME VECTOR
TS - SAMPLE INTERVAL

A.3.12 SUBROUTINE CONTR4

Purpose: Self tuning controller subroutine, converts the routine CONTRL (STC) to the conventional DYFLO from.

Use: CALL CONTR4(VI,CO,ZR,FSRDG,SP,AXN,PB,TI,DER,TS)

Parameters:

VI - CONTROLLER INPUT VECTOR
CO - CONTROLLER OUTPUT VECTOR
ZR - CONTROLLER ZERO SCALE READING VECTOR
FSRDG - CONTROLLER FULL SCALE READING VECTOR
SP - SETPOINT VECTOR
AXN - CONTROLLER ACTION VECTOR
PB - PROGORTIONAL BAND VECTOR
TI - INTEGRAL TIME VECTOR
DER - DERIVATIVE TIME VECTOR
TS - SAMPLE INTERVAL

A.4 Labeled COMMON Block Description

A.4.1 COMMON/AREA9/

Purpose: Carries self-tuning controller constants and data, initialized in routine MIMOIN usually by READing STC.D. Some other initialization is done in the routine CNTR4I.

```
COMMON /AREA9/ IPIS(2), SAVE(2,12,30), YN(2), DNPR(2),
 USS(2), NP(2), PN(2,4), PD(2,4), PHI(2), ACLOSS(2),
 ITYID(2), NSELF(2), N(2,5), KDEL(2,3), ITYCNT(2),
 YPRD(2), SCO(2), SCA(2), SCB(2), NQ(2), QD(2,4), QN(2,4),
 UNT(2), UTL(2), UBL(2), IRAMP(2), UN(2), DS(2,64), NR(2),
 RN(2,4), RD(2,4), WO(2), WN(2), DN(2), ACT(2)
```

Parameters:

IPIS - pointer (counter) for SAVE data structure
SAVE - data structure to hold self tuning data
YN - controlled variables
DNPR - predicted disturbance
USS - steady state controller output
NP - number of "P's"
PN - P polynomial numerator (4 terms max.)
PD - P polynomial denominator (4 terms max.)
PHI - auxillary output function (not used)
ACLOSS - accumulated loss (SP-YN)
ITYID - type of identification 1 ---> RLS
2 ---> RSR

3 ---> RUD
4 ---> RLM

NSELF - total number of parameters/loop (i.e. #G1+#F+#L+#H+#G2 terms)
N - number of G1, F, L, H, G2 parameters
KDEL - assumed model delays
ITYCNT - type of controller (1=STC)
YPRD - predicted output
SCO - coefficient in solution to equations
SCA - coefficient in solution to equations
SCB - coefficient in solution to equations
NQ - number of terms in the q-weighting polynomial denominator
QD - q polynomial denominator (4 terms max)
QN - q polynomial numerator (4 terms max.)
UNT - unbounded STC control output
UTL - upper control output limit (%)
UBL - lower control output limit (%)
IRAMP - max % change in control output per control interval
UN - bounded control output
DS - auxilliary output function (not used ?)
NR - number of terms in the r-weighting polynomial denominator
RN - r polynomial numerator (4 terms max.)
RD - r polynomial denominator (4 terms max.)
WO - setpoints for use in the self tuning algorithms (%)
WN - setpoints * r polynomial
DN - disturbance input to self tuning ID
ACT - controller action switch direct ---> -1.
reverse ---> 1.

A.4.2 COMMON/AREA10/

Purpose: Carries self-tuning controller logic control flags, inititialized in MIMOIN.

```
 COMMON /AREA10/ IWIO, M, IMSELF(2), IMST(2), NSYS, IPRINT,
      IDNPR(2), IPMAT, IMSTI(2)
```

Parameters:

IWIO - logical unit number for printing
M - dummy to hold STC on/off flag
IMSELF - STC on/off flag (1=STC)
IMST - self tuning control start time (# control intervals)
NSYS - number of control loops in the system
IPRINT - print interval for STC data and statistics
IDNPR - type of disturbance measurement none ---> 0
estimated --->1-3
measured ---> 4

IPMAT - covariance matrix print interval
IMSTI - identification start time (# control intevals)

A.4.3 COMMON/CID/

Purpose: Carries parameter identification data, initialized in routine MIMOIN usually by READing STC.D.

```
 COMMON /CID/ X(2,30),PAR(2,30),P(2,465),G(2,30),ROW(2,3),
      IFP(2,3)
```

Parameters:

X - I/O (regressor) vector
PAR - parameter estimate vector
P - covariance matrix
G - kalman gain vector
ROW - forgetting factor vector
IFP - identificatoin on/off flag (0 for ID)

A.4.4 COMMON/CINT/

Purpose: Carries DYFLO control variables, initialized by BLOCK DATA program object code in the file BDATA.2.

```
 COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCO
```

Parameters:
T - simulation time
DT - integration interval
JS - integration control variable
JN - integration control variable
XA - intermediate integration results storage
DXA - intermediate integration results storage
IO - integration order
JS4 - integration control variable
NC - number of components
NCO - index for component number

A.4.5 COMMON/COMP1/

Purpose: Carries loop 1 decoupler constants, initialized interactively by the user at run time.

```
 COMMON/COMP1/KLL1,TLL11,TLL12,TDLL1
```

Parameters:

KLL1 - loop 1 lead lag compensator gain
TLL11 - lag time constant in Gc12 (minutes)
TLL12 - lead time constant in Gc12 (minutes)
TDLL1 - time delay for Gc12 (minutes)

A.4.6 COMMON/COMP2/

Purpose: Carries loop 2 decoupler constants, initialized interactively by the user at run time.

```
 COMMON/COMP2/KLL2,TLL22,TLL21,TDLL2
```

Parameters:

KLL2 - loop 2 lead lag compensator gain
TLL21 - lead time constant in Gc21 (minutes)
TLL22 - lag time constant in Gc21 (minutes)
TDLL2 - time delay for Gc21 (minutes)

A.4.7 COMMON/CO2BY2/

Purpose: Carries controller constants for two loop control system, initialized interactively at run time.

```
 COMMON/CO2BY2/ZR1,ZR2,FSRDG1,FSRDG2,AXN1,AXN2,PB1,
+PB2,RPT1,RPT2,RT1,RT2,RA1,RA2,OI1,OI2,ODI1,ODI2
```

Parameters:

ZR1 - loop 1 zero reading
ZR2 - loop 2 zero reading
FSRDG1 - loop 1 full scale reading
FSRDG2 - loop 2 full scale reading
AXN1 - loop 1 controller action
AXN2 - loop 2 controller action
PB1 - loop 1 proportional band (100/KC1)
PB2 - loop 2 proportional band (100/KC2)
RPT1 - loop 1 repeat rate (minutes**-1)
RPT2 - loop 2 repeat rate (minutes**-1)
RT1 - loop 1 rate time (minutes)
RT2 - loop 2 rate time (minutes)
RA1 - loop 1 rate amplitude
RA2 - loop 2 rate amplitude
OI1 - loop 1 controller integral action output
OI2 - loop 2 controller integral action output
ODI1 - loop 1 controller derivative action output
ODI2 - loop 2 controller derivative action output

A.4.8 COMMON/DSTBNC/

Purpose: Carries disturbance data, initialized interactively by the user at run time.

```
 COMMON/DSTBNC/DSTEP,DTYPE,DISTT
```

Parameters:

DSTEP - disturbance step magnitude (% or fraction)
DTYPE - disturbance type (1 - s.p. 1 1; 2 - s.p. 1 2; 3 - load)
DISTT - hardness disturbance time (minutes)

A.4.9 COMMON/EXCHNG/

Purpose: Carries nonlinear model data structures, initialized by READing the file MINSIM.D in the routine NLMOD.

```
 COMMON/EXCHNG/STREAM(14,25),EQUIP(5,50),HOLDUP(4,25),
1DATA(5,20),NSIZE,PLUG(9,20,25)
```

Parameters:

STREAM - see data structure notes
EQUIP - see data structure notes
HOLDUP - see data structure notes
DATA - see data structure notes
NSIZE - number of size classes
PLUG - see data structure notes

NOTE: See DATA STRUCTURE NOTES for additional details on these variables

A.4.10 COMMON/IN2BY2/

Purpose: Carries 2 by 2 model constants and initial conditions, initialized by routine LDLMOD and required by routine I2BY2.

```
 COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
```

Parameters:

KL - linear model load t.f. gain vector
KP - linear model process t.f. gain matrix
TAU - linear model process t.f. time constant matrix (minutes)
TAUL - linear model load t.f. time constant vector (minutes)
NTL - integral number of integration intervals in linear model process t.f. time delay matrix

(i.e. time delays/RINT)
NTLL - integral number of integration intervals in linear model load t.f. time delay vector (i.e. time delays/RINT)
DX - linear model process t.f. input vector
DY - linear model response vector incl. load
DD - linear model load t.f. input vector

A.4.11 COMMON/LDLIN/

Purpose: Carries linear model input output data, initialized by routine LDLMOD and changes during run.

```
COMMON/LDLIN/ FOF,SWF,CYOS,SSCYOS,SPL,SSSPL,HRD
```

Parameters:

FOF - linear model fresh ore feed rate (tonnes/h)
SWF - linear model sump water feed rate (tonnes/h)
CYOS - linear model cyclone overflow size (%-100 mesh)
SSCYOS - steady state cyclone overflow size (%-100 mesh)
SPL - linear model sump pulp level (metres)
SSSPL - steady state sump pulp level (m)
HRD - linear model ore hardness (%)

A.4.12 COMMON/LOOP1/

Purpose: Carries loop 1 controller constants and setpoint disturbance data, initialized interactively by the user at run time.

```
COMMON/LOOP1/KC1,TI1,TD1,SP1,CSP1,DIST1,TS1
```

Parameters:

KC1 - loop 1 proportional controller gain
TI1 - loop 1 controller integral time (minutes)
TD1 - loop 1 controller derivative time (minutes)
SP1 - loop 1 setpoint (%-100 mesh)
CSP1 - loop 1 setpoint step magnitude
DIST1 - loop 1 setpoint disturbance time (minutes)
TS1 - loop 1 sample interval (minutes)

A.4.13 COMMON/LOOP2/

Purpose: Carries loop 2 controller constants and setpoint disturbance data, initialized interactively by the user at run time.

```
COMMON/LOOP2/KC2,TI2,TD2,SP2,CSP2,DIST2,TS2
```

Paramters:

KC2 - loop 2 proportional controller gain
TI2 - loop 2 controller integral time (minutes)
TD2 - loop 2 controller derivative time (minutes)
SP2 - loop 2 setpoint (metres)
CSP2 - loop 2 setpoint step magnitude
DIST2 - loop 2 setpoint disturbance time (minutes)
TS2 - loop 2 sample interval (minutes)

A.4.14 COMMON/SIM/

Purpose: Carries the simulation control variables, initialized interactively by the user at run time.

```
 COMMON/SIM/SIMT,RINT,PT,TIM
```

Parameters:

SIMT - total simulation time (minutes)
RINT - integration interval (minutes)
PT - results print out interval (minutes)
TIM - simulation time counter (minutes)

A.4.15 COMMON/STC/

Purpose: Carries self-tuning controller data, initialized by READing the file STC.D in routine MIMOIN (from the STC library) and by the routine CNTR4I.

```
 COMMON/STC/IMODE,ITIM(2),ICONT,IDEL(2),DCG(2,2),D(2),
+SIAE(4),ER(2),ERR1(2),ERR2(2),KCNT(2),PK(2),U1,U2,
+MSELF1,MSELF2
```

Parameters:

IMODE - controller mode 1 ---> P
2 ---> PI
3 ---> PID
ITIM - integer sample time in terms of integration time step size
ICONT - open/closed loop flag 1 ---> loop 1 closed
2 ---> loop 2 closed
3 ---> both loops closed
IDEL - analysis delay (# of integration intervals)
DCG - static decoupling gain matrix
D - disturbance vector
SIAE - sum of the absolute errors
ER - current error signal
ERR1 - error signal one control interval previous
ERR2 - error signal two control intervals previous

KCNT - sampling time counter
PK - controller proportional gain vector
U1 - loop 1 controller output
U2 - loop 2 controller output
MSELF1 - self tuning control on/off flag for loop 1
MSELF2 - self tuning control on/off flag for loop 2

A.5 Data Structure Notes

(1)STREAM(IN,IC): THIS ARRAY CARRIES THE ATTRIBUTES OF THE FLOW STREAMS IN THE GRINDING CIRCIUT.

IN=THE STREAM IDENTIFICATION NUMBER

IC=THE ATTRIBUTE INDEX, WHERE;
1<=IC<=20 IS RESERVED FOR SIZE FREQUENCY INFORMATION
IC=21 THE SOLIDS MASS FLOWRATE
IC=22 THE WATER MASS FLOWRATE
IC=23 THE SLURRY VOLUMETRIC FLOWRATE
IC=24 THE PERCENT SOLIDS IN THE SLURRY

(2)EQUIP(IN,IP): THIS ARRAY CARRIES THE PARAMATERS WHICH SPECIFY THE EQUIPMENT.

IN=THE EQUIPMENT NUMBER

(A) FOR GRINDING MILLS

IP=THE PARAMETER INDEX
1<=IP<=20 IS RESERVED FOR THE FIRST COLUMN OF THE NORMALIZED BREAKAGE MATRIX
21<=IP<=40 IS RESERVED FOR THE FIRST ORDER SELECTION CONSTANTS

IP=41 THE PULP VOLUMETRIC HOLDUP OF THE MILL

(B) FOR CLASSIFIERS

IP=J THE J 'TH COEFFICIENT IN THE CYCLONE MODEL

(C) FOR BLENDERS

NO SPECIFICATIONS NECESSARY

(D) FOR SUMP-PUMP

IP=1 CROSS SECTIONAL AREA OF SUMP
IP=2 SUMP HEIGHT
IP=3 INTERNAL FLAG FOR SUMP OVERFLOW/

```
          EMPTY
     IP=4 COEFFICIENT FOR QUADRATIC RELATING
          DYNAMIC HEAD TO FLOW
     IP=5 SAME AS 4
     IP=6 SAME AS 4
          (I.E TDH=A0 + A1*Q + A2*Q**2)

(3)DATA(IN,ID): THIS ARRAY CARRIES AUXILLIARY
                INFORMATION FOR GLOBAL PROGRAM USAGE.

      IN=1 AND 1<=ID<=20 IS RESERVED FOR THE
           CHARACTERISTIC SIZE INFORMATION

      IN=2 AND ID=1 SOLIDS DENSITY
               ID=2 WATER DENSITY

(4)PLUG(IN,IP,IC): THIS ARRAY CARRIES THE THE
   ATTRIBUTES OF THE FLOWING MATERIAL WITHIN A
   TIME DELAY UNIT (EG. A PIPELINE OR CONVEYOR)

      IN=UNIT NUMBER

      IP=INTERNAL COUNTER FOR HANDLING
        INPUT/OUTPUT

      IC=SAME AS 'STREAM' (SEE (1) ABOVE)

 (5)HOLDUP(IN,IC): THIS ARRAY CARRIES THE ATTRIBUTES
    OF SYSTEM HOLDUPS (EG. SUMP)

      IN=UNIT NUMBER

      IC=1 THRU 20 SAME AS 'STREAM'
      IC=21 SOLIDS MASS HOLDUP
      IC=22 WATER MASS HOLDUP
      IC=23 VOLUME HOLDUP
      IC=24 PERCENT SOLIDS IN SLURRY
      IC=25 SPECIAL ATTRIBUTE EG. PULP
            LEVEL IN SUMP
```

Appendix B: DYFLO2.LIB Subroutine Verification

B.1 Coupled ODE System Software, Subroutine INTCPD

The purpose of this section is to document the software that has been developed to integrate systems of highly coupled ordinary differential equations (ODE's). The software is designed to supplement Franks' DYFLO2 library. In fact, this new software requires the support of DYFLO2! Specifically, INTI, INTSTF, and GJR are needed.

The basic alogorithm and supporting mathematics used have been described elsewhere. Essentially, the procedure uses an implicit integration technique in conjunction with a Newton-Raphson iteration. An additional feature is the inclusion of a "smart" relaxation factor. This "smart" algorithm allows one to speed the convergence of the Newton - Raphson iteration by limiting the magnitude of change in the dependent variable. This implies a knowledge of the system being solved. Thus, the major attraction of this algorithm is when the same system is being solved again and again. The relaxation factor can be "tuned" to give optimal performance.

As alluded to earlier, the software follows the format used in the DYFLO2 library closely. It should be noted that a numerical Jacobian calculation routine is used. Although this is not the best method for computing the Jacobian, this routine enhances the generality of the software. The problem specific code is limited to the two routines used to calculate the forcing function vector (FFCALC) and the inverse time constant vector (TKCALC).

The reader is refered to the source code listing following for further details on implementation. Given the comments internal to the listing and the previous discussion of the integration algorithm, the software is self documenting. Included in the listing is a test driver program for the example problem discussed next. The example is intended to help clarify the use of this software in problem solving.

Example

This example problem has been taken from Rainville and Bedient (1974) as they give an analytical solution. As well, the analytical solution involves exponentials. This type of nonlinearity is well known to cause problems in numerical solutions.

Consider the following set of two first order ordinary differential equations:

$$\frac{dx}{dt} = y \tag{B.1}$$

$$\frac{dy}{dt} = -2x + 3y \tag{B.2}$$

The analytical solution to this system has the form:

$$x=c_1e^t \tag{B.3}$$

$$y=c_1e^t \tag{B.4}$$

or:

$x=c_1e^{2t}$ (B.5)

$y=2c_1e^{2t}$ (B.6)

where c1 is an arbitrary constant.

Some valid initial conditions are:

$x(t_0)=1$ (B.7)

$y(t_0)=2$ (B.8)

Thus, using the second analytical solution, c1=1. Equations (B.5) and (B.6) can be rewritten as:

$x=e^{2t}$ (B.9)

$y=2e^{2t}$ (B.10)

To solve the system numerically using the DYFLO2 software, equations (B.1) and (B.2) must first be written in the standard form. That is:

F1=FF1 - TK1 *X1 (B.11)

F2=FF2 - TK2 *X2 (B.12)

where:

X1=x; X2=y (B.13)

FF1=y; FF2=-2x (B.14)

TK1=0; TK2=-3 (B.15)

In terms of the symbols ~~in~~ the user written subroutines for the forcing function and the inverse time constant calculations, equations (B.14) and (B.15) become:

FFVECT(1)=X(2) (B.16)

FFVECT(2)=-2.*X(1) (B.17)

and:

TKVECT(1)=0 (B.18)

TKVECT(2)=-3 (B.19)

These last 2 pairs or equations are coded into the subroutines FFCALC and TKCALC, respectively, as they are given here.

It remains only to code the executive program to carry out the desired calculations as is standard when using DYFLO2. A driver for this problem is given in the source code listing in Figure B.1. The analytical solution given by equations (B.9) and (B.10) has been included in the program for comparison with the numerical solution as well. Note the simplicity of the driver.

The numerical and analytical solutions using an integration step of 0.1 time units are contrasted in Figure B.2. Note that the numerical solution for both dependent variables is always larger than the analytical solution. This type of behavior is noted by Franks as well. Figure B.3 is a similar plot using an integration step of 0.05 time units. Again, the numerical solution is larger than the analytical solution. The salient feature of this plot however, is the improvement in accuracy over that indicated by Figure B.2. Obviously, this is expected as a smaller integration step was used to produce Figure B.3.

Attempts to use time steps larger than 0.1 units lead to disasterous results. The linear system of equations from the Newton-Raphson iteration became singular and so the Gauss-Jordan Reduction routine (GJR) failed. This is thought to be due to round off errors within the computer. As DYFLO2 is written using single precision floating point

```
C***********************************************************
C*  COUPLE SYSTEM SOFTWARE TEST DRIVER                     *
C*                                                         *
C*   DYFLO2 SIMULATION      85/08/01                       *
C***********************************************************
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4
      REAL X(30),XACT(2),W(30),CMAX(30),CMIN(30)
      LOGICAL NF
C
C   INITIALIZATION
C
      TIM=0.
      N=2
      X(1)=1.
      X(2)=2.
      RINT=.1
      SIMT=3.
      IO=1
      JN=0
      JS=0
      JS4=0
      NF=.FALSE.
      PT=0.1
      CMAX(1)=.1
      CMAX(2)=.1
      CMIN(1)=0.00001
      CMIN(2)=0.00001
      W(1)=1.
      W(2)=1.
C
C   DERIVATIVE SECTION
C
C ANALYTICAL SOLUTION
C
    10 XACT(1)=EXP(2.*TIM)
       XACT(2)=2.*EXP(2.*TIM)
C
C   TEST FOR PRINT AND FINISH
C
       CALL PRNTF(PT,SIMT,NF,TIM,0.,X(1),X(2),0.,XACT(1),XACT(2),
     +0.,W(1),W(2))
       IF(NF) GOTO 20
C
C   INTEGRATION SECTION
C
       CALL INTI(TIM,RINT,IO)
       CALL INTCPD(N,CMAX,CMIN,X,W)
       GOTO 10
C
C   TERMINATION SECTION
C
    20 STOP
       END
```

Figure B.1: Coupled System DYFLO2 Executive Program Source Code Listing

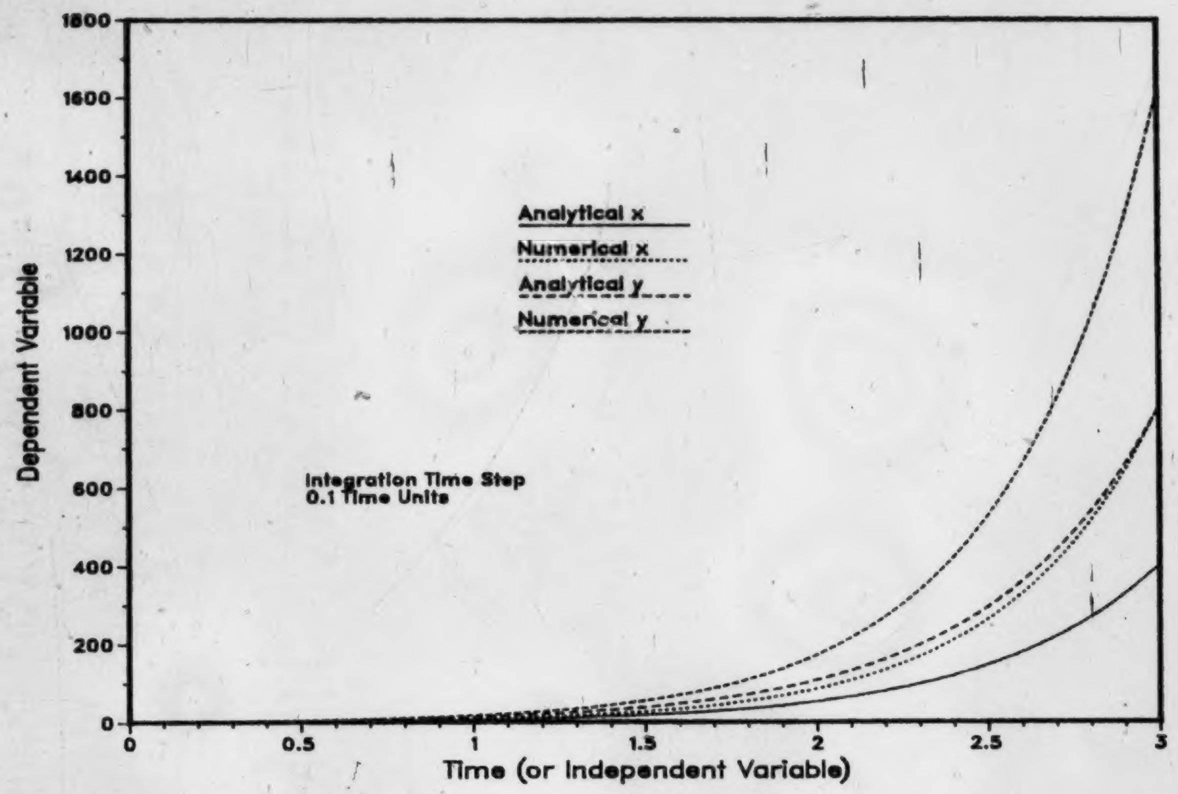


Figure B.2: Analytical versus Numerical Solution Using the Iterative Implicit Integration Technique for an Integration Time Step of 0.1 Units

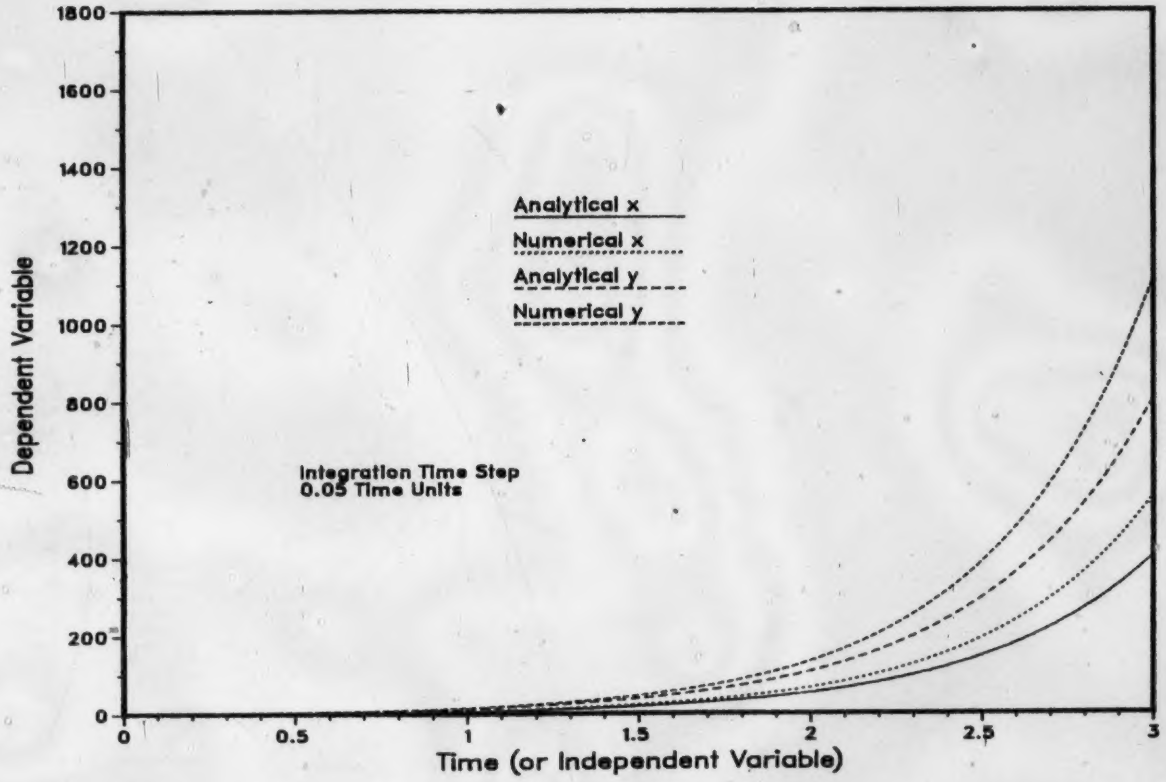


Figure B.3: Analytical versus Numerical Solution Using the Iterative Implicit Integration Technique for an Integration Time Step of 0.05 Units

arithmetic, so were the new integration routines discussed here. A more robust software system would result if double precision were used. Noticable improvement however, is doubtful due to the nature of the integration scheme. A better place to start is a more advanced implicit integration algorithm.

B.2 Gauss-Jordan Reduction Subroutine GJR

As previouse use of the subroutine GJR (Gauss Jordan reduction) has not been documented at the University of Alberta, testing was deemed necessary before using it in the coupled ODE routine, INTCPD.

The following simple 2x2 system of equations was used for the test:

$$2x_1 + 3x_2 = 12 \tag{B.20}$$

$$5x_1 + 6x_2 = 27 \tag{B.21}$$

The solution is quickly found "by hand". Subtracting twice the first equation form the second yields immediately:

$$x_1 = 3 \tag{B.22}$$

Substituting this value back into the first equation gives:

$$x_2 = 2 \tag{B.23}$$

It remains to check this analytical solution with the one obtained numerically by GJR. Note that although it has not been formally stated, the Gauss-Jordan solution procedure was used in the above solution. A further explanation of the technique can be found in any introductory numerical analysis text.

To use GJR, the system of equations must be written in matrix form as follows:

$$\mathbf{Ax} = \mathbf{b} \tag{B.24}$$

Converting equations (B.20) and (B.21) to this notation gives:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix} \tag{B.25}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{B.26}$$

$$\mathbf{b} = \begin{bmatrix} 12 \\ 27 \end{bmatrix} \tag{B.27}$$

Furthermore, **A** must be augmented with **b** to conform to the input structure required by GJR. Thus:

$$\mathbf{A'} = \left[\begin{array}{cc|c} 2 & 3 & 12 \\ 5 & 6 & 27 \end{array}\right] \tag{B.28}$$

Figure B.4 gives a source code listing of a test program that solves the example system discussed here. The solution produced by this program is found in Figure B.5. If the number of decimal places used in the solution output is considered, it is clear that the numerical solution is accurate.

It should be noted that one additional feature of GJR is that it computes the determinant of **A**. To check the correctness of this calculation, the analytical determinant of equation (B.25) is:

$$2(6) - 3(5) = -3 \tag{B.29}$$

```
C   DYFLO2 SUBROUTINE GJR TEST DRIVER PROGRAM
      REAL A(30,30),X(30)
      A(1,1)=2.
      A(1,2)=3.
      A(1,3)=12.
      A(2,1)=5.
      A(2,2)=6.
      A(2,3)=27.
      N=2
      TOL=1.E-06
      KEY=0
      CALL GJR(A,N,TOL,KEY,X,DETER)
      WRITE(6,1) DETER
    1 FORMAT(/,'    DETERMINANT = ',F12.6)
      WRITE(6,2) X(1),X(2)
    2 FORMAT(2F12.6)
      STOP
      END
```

Figure B.4: Subroutine GJR Test Executive Program Listing

```

    DETERMINANT =    -2.999994
      3.000006    1.999996
```

Figure B.5: Sample Output from Subroutine GJR Test Program

Again, the numerical determinant in Figure B.5 is accurate.

B.3 Zero Order Hold Subroutine ZOH

The Laplace transform of a zero order hold is defined as:

$$ZOH = \frac{1 - e^{-T_s s}}{s} \qquad (B.30)$$

where T_s is the sample interval. The inverse Laplace transform of equation (B.30) is:

$$\frac{dy(t)}{dt} = x(t) - x(t-T_s)u(t-T_s) \qquad (B.31)$$

With this information, it is a simple matter to develop a DYFLO2 compatable subroutine to perform the duties of a zero order hold. The FORTRAN source code for this routine can be found in Appendix D along with a brief explanation of the logic involved. A short DYFLO2 test executive program is given in Figure B.6 and sample output is given in Figure B.7. The input function to the ZOH is a sine wave with

```
C
C   ZERO ORDER HOLD TEST DRIVER
C
C   INPUT IS A CONTINUOUS SINE WAVE
C   SAMPLE INTERVAL IS 5 "DO LOOP" ITERATIONS
      TIM=0.
      A=0.
      C=2.*3.14159265/100.
      WRITE(6,2)
    2 FORMAT(//,4X,'TIME',8X,'INPUT',7X,'OUTPUT',/)
      DO 10 I=1,100
      A=A+C
      B=SIN(A)
      CALL ZOH(OUT,B,5.,TIM,1)
      WRITE(6,1) TIM,B,OUT
    1 FORMAT(3F12.6)
   10 TIM=TIM+1.
      STOP
      END
```

Figure B.6: Subroutine ZOH Text Executive Program Listing

```
   TIME          INPUT        OUTPUT

   0.0          0.062790     0.062790
   1.000000     0.125333     0.062790
   2.000000     0.187381     0.062790
   3.000000     0.248690     0.062790
   4.000000     0.309017     0.062790
   5.000000     0.368124     0.368124
   6.000000     0.425779     0.368124
   7.000000     0.481753     0.368124
   8.000000     0.535826     0.368124
   9.000000     0.587785     0.368124
  10.000000     0.637424     0.637424
  11.000000     0.684547     0.637424
  12.000000     0.728968     0.637424
  13.000000     0.770513     0.637424
  14.000000     0.809017     0.637424
  15.000000     0.844328     0.844328
  16.000000     0.876306     0.844328
  17.000000     0.904826     0.844328
  18.000000     0.929775     0.844328
  19.000000     0.951055     0.844328
  20.000000     0.968582     0.968582
  21.000000     0.982286     0.968582
  22.000000     0.992114     0.968582
  23.000000     0.998026     0.968582
  24.000000     1.000000     0.968582
  25.000000     0.998027     0.998027
  26.000000     0.992116     0.998027
  27.000000     0.982289     0.998027
  28.000000     0.968586     0.998027
  29.000000     0.951061     0.998027
  30.000000     0.929782     0.929782
  31.000000     0.904834     0.929782
  32.000000     0.876315     0.929782
  33.000000     0.844337     0.929782
  34.000000     0.809028     0.929782
  35.000000     0.770526     0.770526
  36.000000     0.728982     0.770526
  37.000000     0.684562     0.770526
  38.000000     0.637441     0.770526
  39.000000     0.587804     0.770526
  40.000000     0.535847     0.535847
  41.000000     0.481775     0.535847
  42.000000     0.425802     0.535847
  43.000000     0.368149     0.535847
  44.000000     0.309043     0.535847
  45.000000     0.248717     0.248717
  46.000000     0.187410     0.248717
  47.000000     0.125363     0.248717
  48.000000     0.062822     0.248717
  49.000000     0.000032     0.248717
  50.000000    -0.062757    -0.062757
```

Figure B.7: Sample Output from Subroutine ZOH Test Program

unity magnitude and a period of 100 time units. The sample interval is 5 time units.

B.4 Fixed Time Delay Subroutine TDL

To model fixed time delays on a digital computer, data must simply be stored for a given period of simulation time before being output. A complete discussion on the theory behind time delay modelling has been given in Chapter 3 and will not be repeated here.

A DYFLO2 (FORTRAN IV) implementation of a subroutine that uses the above time delay modelling technique is shown in Figure B.8. Brief parameter descriptions are given in the source code with a more detailed description presented in Table B.1. All variables internal to the routine are described in Table B.2.

The routine is designed to function with all DYFLO2 integration routines. By passing the variables YA (work area or circular buffer matrix) and NTLC (time delay location counter) through the parameter list, the time delay routine has been made local to the calling program. The dimension of NTLC indicates the number of buffers and thus the number of distinct time delays available for use in a given program. This dimension is the same as the first dimension of the work area matrix, YA. The second dimension of YA sets the maximum value that NTL (number of time locations or ratio of the time delay to the integration time step size) can take. The actual value of NTL determines how many of the locations are used to simulate a particular time

```
C ********** TDL **********
C
C TIME DELAY SUBROUTINE
C
C PARAMETERS: YA      -WORK AREA ARRAY
C             NTLC    -TIME DELAY LOCATION COUNTER
C             Y       -INPUT SIGNAL
C             YL      -OUTPUT DELAY SIGNAL
C             NTL     -NUMBER OF TIME LOCATIONS IN TIME DELAY
C                       (RATIO OF TIME DELAY TO INTEGRATION STEP SIZE)
C             JC      -SUBROUTINE CALL NUMBER
C
C***NOTE:  NTLC MUST BE INITIALIZED TO 25*0 BEFORE THIS
C          SUBROUTINE IS CALLED BY THE DECLARATION:
C                      INTEGER NTLC(25)/25*0/
C
      SUBROUTINE TDL(YA,NTLC,Y,YL,NTL,JC)
      COMMON/CINT/ T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      DIMENSION YA(25,500),NTLC(25)
      JNTL=NTL
      IF(IO .EQ. 4) JNTL=JNTL+1
      IF (NTLC(JC) .LE. 0) GO TO 3
      GO TO (5,6,6,8),IO
    6 GO TO (5,7),JS
    3 DO 4 K = 1,JNTL
    4 YA(JC,K) = Y
    5 NTLC(JC) = NTLC(JC)+1
      IF (NTLC(JC) .GT. JNTL) NTLC(JC)=1
    7 NCJ=NTLC(JC)
      YL = YA(JC,NCJ)
      IF (JS .EQ. 2) YA(JC,NCJ)=Y
   10 RETURN
    8 GO TO (9,10,5,11),JS4
    9 NCJ=NTLC(JC)
      NC1=NCJ+1
      IF (NC1 .GT. JNTL) NC1 = 1
      YL = (YA(JC,NCJ)+YA(JC,NC1))/2.
      RETURN
   11 NCJ=NTLC(JC)-1
      IF(NCJ.EQ.0) NCJ=JNTL
      YA(JC,NCJ)=Y
      RETURN
      END
```

Figure B.8: DYFLO2 Fixed Time Delay Subroutine TDL Listing

Table B.1: Parameter Description for the DYFLO2 Routine TDL

YA - subroutine work area; the circular buffers are stored as rows in this matrix. The variable JC (subroutine call number) is used to index the particular buffer (row) to be used and the variable NTLC is used to index the particular cell (column) element to be output next.

NTLC - time delay location counter; this vector carries the index of the particular cell (column) in the circular buffer that is to be output from the current subroutine call. The variable JC (subroutine call number) is used to index the particular buffer that is to be used as this counter is specific to each buffer.

Y - input signal; variable that is to be delayed by NTL integration time steps.

YL - delayed output signal; this is the variable Y delayed by NTL integration time steps.

NTL - number of time locations (in the time delay); an integer number that corresponds to the total number of integration time steps in the time delay. This variable is calculated as the ratio of the time delay to the integration time step size.

JC - subroutine call number; indexes the appropriate circular buffer to be used. Both YA and NTLC use this index.

Notes:

1. NTLC must be initialized to 25 * 0 in the calling routine by the declaration:
   `INTEGER NTLC/25*0/`

2. This routine can be used up to 25 times (i.e. the maximum JC value is 25) in a single calling routine. In other words, this routine is local to the calling program.

3. The second dimension on the variable YA implies that NTL can at most be equal to 500.

Table B.2: Internal Variable Description for the DYFLO2 Routine TDL

IO - integration order; located in the COMMON/CINT/ and used in this routine for logic control.

JNCL - dummy number of time locations in the dime delay; this variable is used to prevent interaction of this routine with the value stored in NTL (as passed from the calling program) when fourth order integration is used.

JS - second order integration control flag; this flag is set by the DYFLO2 routine INTI and is used in this routine to indicate when a "legitimate" input variable is available for storing into YA. That is, because two passes through the INTEGRATION SECTION of the DYFLO2 program are required to complete the second order integration procedure, only the second values computed for the dependent variables are valid.

JS4 - fourth order integration control flag; this flag is set by the DYFLO2 routine INTI and is used in this routine to indicate when a "legitimate" input variable is available for storing into YA. It is similar in function to JS, except that four passes through the INTEGRATION SECTION of the DYFLO2 program are required to complete a single integration step.

NCJ - time delay location counter dummy index to ease coding.

NC1 - time delay loaction counter plus 1; this variable is used to index the next buffer cell when the fourth order integration procedure is used and is needed for the calculation of the output variable for the two intermediate passes. That is, the output variable for these two passes is the average of the two legitimate values at either end of the current time step.

delay. The subroutine logic is summarized in Table B.3.

This subroutine may be placed in either the derivative or integration sections of the DYFLO2 executive program. Normally, if the time delay is associated with a dynamic element it should appear in the integration section with the dynamic model as this will help to maintain program readability. The example that follows will clarify this feature.

TDL Example

Problem Statement:

A simple plant, modelled by two first order plus time delay (FOPTD) transfer functions, is shown in the block diagram in Figure B.9. Simulate this process in terms of deviation variables for 10 minutes using DYFLO2. Excite the system with a unit step in the input variable after one

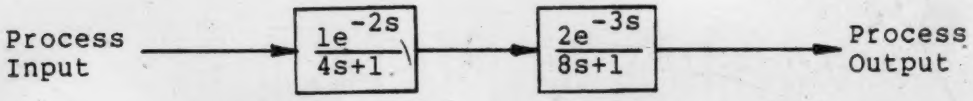


Figure B.9: Block Diagram Representation of a Simple Process

Table B.3: Summary of TDL (DYFLO2) Logic

Line Numbers	Comments
1 to 15	comments, internal documentation
16	SUBROUTINE definition line with the parameter list
17	Integartion control COMMON definition, included here because the integration control flags, JS and JS4, and the integration order flag are required by this routine
18	DIMENSION statement
19	Sets dummy number of time locations
20	IF fourth order integration is being used, add one extra cell to the buffer
21	IF statement for initialization of time delay buffer
22	GO TO appropriate location in subroutine for the specified integration order
23	Second order integration control, GO TO the appropriate location in subroutine based on the integration pass as indicated by the value of the flag, JS
24 to 25	DO loop for buffer initialization, the first call to this routine loads each cell of the buffer with the initial value of the input signal.

26	Increments the time delay location counter (rotates the buffer) to index current output cell. The subscript, JC, on NTLC references the particular buffer. This line is the entry point for first order integration, the first pass of second order integration, and the third pass of fourth order integration.
27	IF NTLC is past the end of the buffer, reset to the first cell.
28	Sets dummy counter for coding ease. This line is the entry point for the second (legitimate) pass for second order integration.
29	Get the (delayed) output signal from the buffer matrix, buffer cell now "empty".
30	Puts the new input signal into the cell emptied by line 29. The variable, JS, has the value 2 when a legitimate integration pass is performed (i.e. every pass for first order integration, but only every second pass with second order integration. JS is not used with fourth order integration).
31	RETURN to calling routine if first or second order integration being used. Entry point for second (dummy) pass of fourth order integration.
32	Fourth order integration control, GO TO the appropriate location in the subroutine based on the integration pass as indicated by the value of the flag, JS4.
33	Sets dummy counter for ease of coding. This line is the entry point for the first pass for fourth order integration. It is a dummy pass.

34	Sets a second dummy counter to index the output cell temporarily for use in the average output calculation (line 36).
35	Resets the dummy counter from line 34 if necessary.
36	Averages the output signal at the start and finish of the integration interval for use with the two dummy passes (JS4 = 1 and 2).
37	RETURN from first pass of fourth order integration.
38	Sets dummy counter for ease of coding. This line is the entry point for the fourth pass of fourth order integration.
39	Checks to make sure dummy counter set in line 38 is in correct range. Resets if necessary.
40	Puts the new input signal into the appropriate cell in the buffer.
41	RETURN from fourth pass of fourth order integration.
42	END of subroutine code.

minute of steady state operation. Perform the simulation using each of the three integration techniques (first, second, and fourth order) in turn. Examine the behavior of the time delay routine in each case.

Solution:

The original block diagram must first be decomposed into seperate operations that DYFLO2 is capable of handling. This results in the block diagram shown in Figure B.10. The routine TFN1 can be used to simulate the two first order transfer functions. The time delays are modelled using TDL and are cascaded with the appropriate transfer function routine.

Figure B.11 is the source code listing of the program used for the simulations. A suitable integration interval is 0.1 minutes. Lines 8 through 43 initialize both the

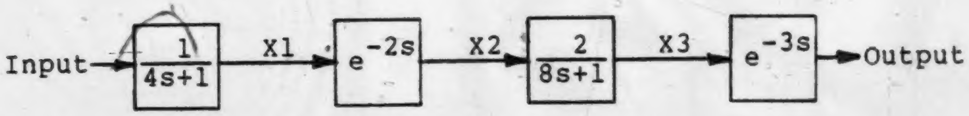


Figure B.10: Block Diagram Decomposed For DYFLO2 Simulation

```
C
C TDL SUBROUTINE TEST PROGRAM
C
      COMMON/CINT/ T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      REAL YA(25,500),K1,K2,INPUT
      INTEGER NTLC(25)/25*0/
      LOGICAL NF
C
C   INITIALIZATION SECTION
C
C *** SYSTEM ***
      IO=1
      JS=0
      JN=0
      JS4=0
      SIMT=10.
      DISTIM=1.
      TIM=0.
      RINT=.1
      PT=RINT
      NF=.FALSE.
C *** PROCESS MODEL ***
      K1=1.
      K2=2.
      T1=4.
      T2=8.
      TD1=2.
      TD2=3.
      NTL1=INT(TD1/RINT)
      NTL2=INT(TD2/RINT)
C *** INITIAL MODEL CONDITIONS ***
      INPUT=0.
      X1=0.
      X2=0.
      X3=0.
      OUTPUT=0.
C *** OUTPUT HEADINGS ***
      WRITE(6,1) IO
    1 FORMAT(///,5X,'TIME DELAY ROUTINE TEST',//,5X,
     +       'INTEGRATION ORDER = ',I3)
      WRITE(6,2)
    2 FORMAT(/,4X,'TIME',19X,'INPUT',9X,'X1',10X,'X2',10X,'X3',
     +       8X,'OUTPUT',/)
C
C   DERIVATIVE SECTION
C
C *** MAIN SIMULATION LOOP ***
   10 IF(TIM .GE. DISTIM) INPUT=1.
      DX=X2
C *** TEST FOR PRINT AND FINISH ***
      CALL PRNTF(PT,SIMT,NF,TIM,0.,INPUT,X1,X2,X3,OUTPUT,0.,0.,0.)
      IF(NF) GOTO 20
C
C   INTEGRATION SECTION
C
      CALL INTI(TIM,RINT,IO)
C *** PLANT MODEL CALCULATIONS ***
C      CALL TFN1(X1,INPUT,T1,K1)
      X1=INPUT
      CALL TDL(YA,NTLC,X1,X2,NTL1,1)
      CALL INTG(X2,DX)
C      CALL TFN1(X3,X2,T2,K2)
      X3=X2
      CALL TDL(YA,NTLC,X3,OUTPUT,NTL2,2)
      GOTO 10
C
C   TERMINATION SECTION
C
   20 STOP
      END
```

Figure B.11; Subroutine TDL Test Executive Program Listing system and model variables. The derivative section of the program (lines 44 through 51) handles the disturbance calculations and tests for printing and simulation termination. The plant model appears in the integration section (lines 52 through 61) because TFN1 is a dynamic model and contains a call to INTG (or INTSTF). It should be noted that the calls to TDL are made in this section because the delays are part of the plant model. Lines 62 through 66 are the termination section of the program.

Results of simulation tests using first, second, and fourth order integration techniques are presented in Tables B.4, B.5, and B.6, respectively. The dummy state variables X1, X2, and X3 are included in these printouts so that the behavior of each individual call to TDL can be examined. In general, the routine produces the expected results. The time from when the disturbance enters the system to when it affects X2 and X3 is two minutes and in turn there is no effect of X3 on the output signal until after the 3 minute delay.

Table B.4: Sample Output for the Fixed Time Delay Simulation using First Order Integration

```
  TIME DELAY ROUTINE TEST

  INTEGRATION ORDER =    1

 TIME                      INPUT        X1           X2           X3           OUTPUT

0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
2.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
3.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
4.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
5.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
6.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
7.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
8.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
9.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.1000E+00   0.0          1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0
3.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1000E+00   1.1000E+00   0.0          0.0          0.0          0.0
3.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1100E+00   1.1100E+00   0.0          0.0          0.0          0.0
3.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1110E+00   1.1110E+00   0.0          0.0          0.0          0.0
3.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
3.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
3.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
3.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
4.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
5.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
6.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
6.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0          0.0
6.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.0000E+00   0.0          0.0          0.0
6.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1000E+00   0.0          0.0          0.0
6.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1100E+00   0.0          0.0          0.0
6.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1110E+00   0.0          0.0          0.0
6.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
6.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
6.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
6.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
7.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
8.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
9.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1111E+00   1.1111E+00   1.1111E+00   0.0          0.0          0.0
```

Table B.5: Sample Output for the Fixed Time Delay Simulation using Second Order Integration

```
TIME DELAY ROUTINE TEST

INTEGRATION ORDER =   2

TIME                       INPUT        X1           X2           X3           OUTPUT

0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
2.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
3.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
4.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
5.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
6.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
7.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
8.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
9.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0500E+00   1.0500E+00   0.0          0.0          0.0          0.0
3.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0027E+00   1.0027E+00   0.0          0.0          0.0          0.0
3.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
3.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
4.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
5.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
6.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0          0.0
6.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0500E+00   0.0          0.0          0.0
6.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0027E+00   0.0          0.0          0.0
6.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
6.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
7.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.8000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
8.9000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.0000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.1000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.3000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.4000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.5000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.6000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.7000E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.7999E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.8999E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
9.9999E+00   0.0          1.0000E+00   1.0000E+00   1.0048E+00   1.0048E+00   1.0048E+00   0.0          0.0          0.0
```

Table B.6: Sample Output for the Fixed Time Delay
Simulation using Fourth Order Integration

```
   TIME DELAY ROUTINE TEST

   INTEGRATION ORDER =    4

  TIME                     INPUT          X1           X2           X3        OUTPUT

0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
2.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
3.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
4.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
5.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
6.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
7.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
8.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
9.0000E-01   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
1.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
1.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.1000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.2000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.3000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.4000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.5000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.6000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.7000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.8000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
2.9000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.0000E+00   0.0          1.0000E+00   1.0000E+00   0.0          0.0          0.0          0.0          0.0          0.0
3.1000E+00   0.0          1.0000E+00   1.0000E+00   8.4337E-01   8.4337E-01   0.0          0.0          0.0          0.0
3.2000E+00   0.0          1.0000E+00   1.0000E+00   1.0868E+00   1.0868E+00   0.0          0.0          0.0          0.0
3.3000E+00   0.0          1.0000E+00   1.0000E+00   1.1070E+00   1.1070E+00   0.0          0.0          0.0          0.0
3.4000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
3.5000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
3.6000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
3.7000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
3.8000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
3.9000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.0000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.1000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.2000E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
4.9999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.0999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.1999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
5.9999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0          0.0
6.0999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   8.4337E-01   0.0          0.0          0.0
6.1999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.0868E+00   0.0          0.0          0.0
6.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1070E+00   0.0          0.0          0.0
6.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
6.9999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.0999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.1999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
7.9999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.0999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.1999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
8.9999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.0999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.1999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.2999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.3999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.4999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.5999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.6999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.7999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
9.8999E+00   0.0          1.0000E+00   1.0000E+00   1.1071E+00   1.1071E+00   1.1071E+00   0.0          0.0          0.0
```

Appendix C: A Review of some Numerical Techniques with Consideration to DYFLO/DYFLO2

One of the most important tasks involved in any simulation of a dynamic system is that of obtaining a solution to the set of ordinary differential equations (ODE's) that describe the physical situation. As the term "simulation" implies the use of a digital computer to carry out computations, the discussion may be immediately restricted to numerical techniques.

The purpose of this appendix is to provide a summary of the theory behind the techniques Franks (1972, 1982) has used to develop the DYFLO and DYFLO2 library routines for solving ODE's. The underlying objective is to present the material in more rigorous mathematical terms than Franks has in his documentation. Readers who are familiar with simple numerical ODE solution techniques may skip to the section on coupled systems if desired. However, it may be found advantageous to skim the earlier sections to pick up on the notation to ease the understanding of the conversion to that of Franks.

Before proceeding further, it should be noted that the phrase, "set of ODE's", has been used in a broad sense meaning "one to many ODE's" throughout this appendix. No loss will be incurred if the matrix equations are interperated as if they were scalar.

C.1 Explicit (Forward) Euler

Consider the typical set of n first order ODE's given by the following matrix equation:

$$\frac{du(t)}{dt} = f(u(t)) \qquad (C.1)$$

with the initial conditions:

$$u_o = u(t_o) \qquad (C.2)$$

It should be noted here that is is assumed that a solution exists. Rewriting equation (C.1) using a forward Taylor series expansion gives:

$$u(t) = u(t_o) + (t-t_o)f(u(t_o)) + \frac{(t-t_o)^2}{2!}f'(u(t_o)) + \frac{(t-t_o)^3}{3!}f''(u(t_o)) + \ldots \qquad (C.3)$$

Substituting the definition:

$$\Delta t = t - t_o \qquad (C.4)$$

into equation (C.3), and then truncating the infinite series after the linear terms yields the approximate solution:

$$u(t) \simeq u(t_o) + \Delta t f(u(t_o)) \qquad (C.5)$$

which is the algorithm for the explicit Euler ODE solution technique. A better understanding of this integration technique can be gained by considering Figure C.1 where it can be seen that the solution is projected forward in time in a linear fashion according to the slope of the function at the current time. Obviously, a larger time step (Δt) reduces the confidence in the accuracy of the solution.

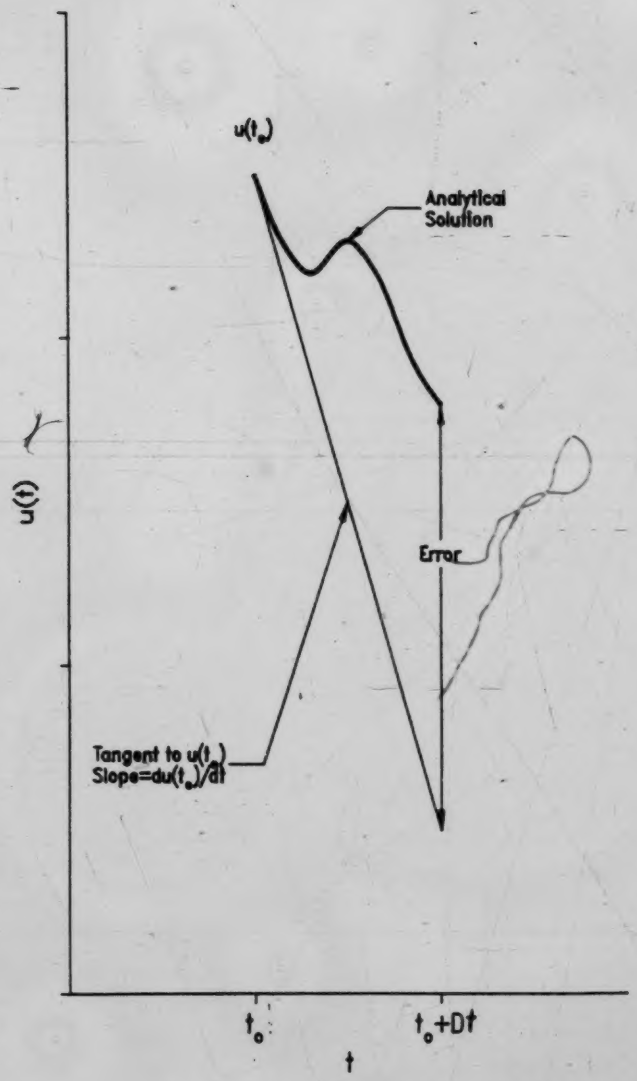

u(t)
$u(t_o)$
Analytical Solution
Error
Tangent to $u(t_o)$
Slope=$du(t_o)/dt$
t_o
t_o+Dt
t

Figure C.1: Mechanics of the Explicit Euler Solution Technique

It is convenient to change the notation used in equation (C.5). By letting:

$$u(t) = u^{k+1} \tag{C.6}$$

$$u(t_o) = u^k \tag{C.7}$$

and then:

$$f(u(t_o)) = f^k \tag{C.8}$$

equation (C.5) becomes:

$$u^{k+1} \simeq u^k + \Delta t f^k \tag{C.9}$$

This change of notation forces the algorithm to advance itself through time in a recursive fashion. In fact, the superscript, k, counts the number of Δt's added to t_o. This notational style is appropriate for implementation on a digital computer and Figure C.2 gives a graphical interpretation of equation (C.9). The solution is advanced along the trajectory tangent that passes through the current solution position.

It should be noted that equation (C.9) may be made to give the exact solution if an error term were to be included. However, the local error is proportional to Δt^2 and so, if the time step is chosen small enough, the error can be neglected for most practical purposes. Usually, solution stability dictates the maximum step size that may be used in a given situation.

C.2 Implicit (Backward) Euler

The development of this method follows from the discussion given for the explicit Euler method. Starting with the same set of ODE's (equations (C.1) and (C.2)),

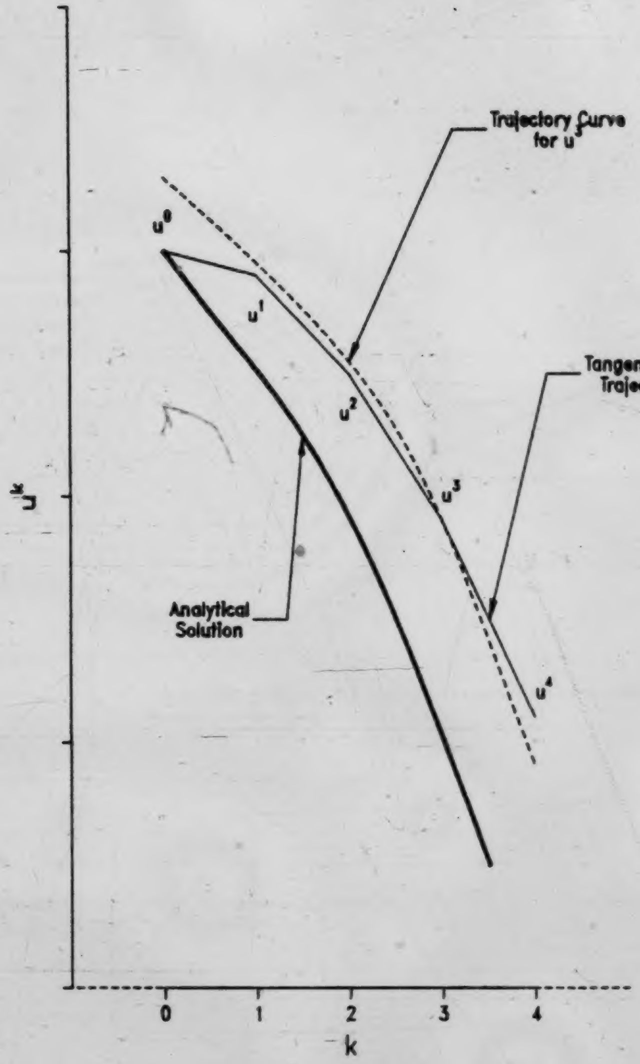


Figure C.2: Graphical Integration of the Recursive Form of the Explicit Euler ODE Solution Technique

equation (C.1) can be expanded using a backward Taylor series:

$$u(t_o)=u(t)-(t-t_o)f(u(t))-\ldots \qquad (C.10)$$

Substituting equation (C.4) into (C.10) and truncating as before yields the approximation:

$$u(t_o)\simeq u(t)-\Delta t f(u(t)) \qquad (C.11)$$

The implicit Euler algorithm follows from rearranging equation (C.11):

$$u(t)\simeq u(t_o)+\Delta t f(u(t)) \qquad (C.12)$$

Incorporating equations (C.6) through (C.8) into equation (C.12) gives the recursive formula:

$$u^{k+1}\simeq u^k+\Delta t f^{k+1} \qquad (C.13)$$

This formula, however, is quite useless because u^{k+1} is required to compute f^{k+1}! Figure C.3 attempts to interpret this equation graphically. It is obvious that u_4 is needed before u_3, and u_3 is needed before u_2, and so on. There are several alternate formulations available that avoid this problem. The one most widely used involves the approximation of f^{k+1} with a truncated Taylor series:

$$f^{k+1}\simeq f^k+(u^{k+1}-u^k)J^k \qquad (C.14)$$

where:

$$J^k=f^{k\prime}=df^k/du^k \qquad (C.15)$$

Equations (C.15) is known as the Jacobian matrix. For those readers unfamiliar with the matrix calculus operations in equation (C.15), the Jacobian for a set of n ODE's is:

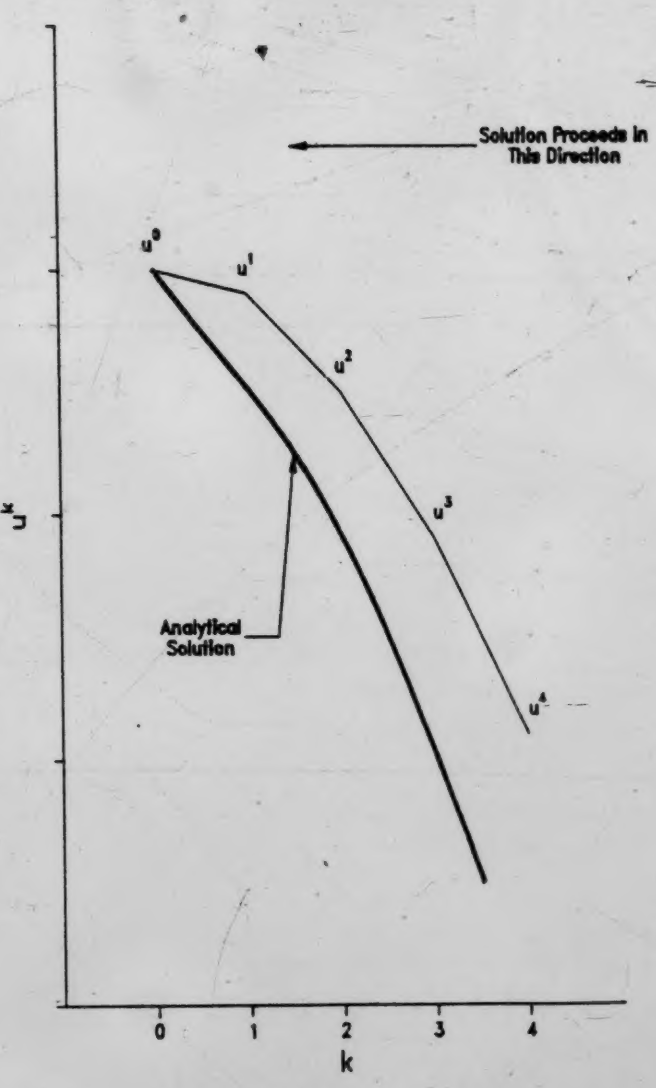


Figure C.3: Graphical Integration of the Recursive Form of the Implicit Euler ODE Solution Technique

$$J = \begin{bmatrix} \frac{df_1}{du_1} & \frac{df_1}{du_2} & \frac{df_1}{du_3} & \cdots & \frac{df_1}{du_n} \\ \frac{df_2}{du_1} & & & & \vdots \\ \frac{df_3}{du_1} & & & & \\ \vdots & & & & \\ \frac{df_n}{du_1} & \cdots & & & \frac{df_n}{du_n} \end{bmatrix} \qquad (C.16)$$

Substituting equation (C.14) into (C.13) gives:

$$u^{k+1}=u^k+\Delta t(f^k+u^{k+1}-u^k)J^k$$

(C.17)

and solving for u^{k+1} gives the following form of the implicit Euler algorithm:

$$u^{k+1}\approx u^k+[I-\Delta tJ^k]^{-1}\Delta tf^k \qquad (C.18)$$

All terms in equation (C.18) are known or may be calculated at time step k, allowing the solution to be advanced through time. The obvious drawback is the fact that the inverse of a matrix is required each integration time step. This is trivial in the case of a single ODE because the Jacobian matrix is simply a scalar. The advantage of this type of scheme is that the time step size (Δt) is not limited. In other words, the solution is stable even for Δt approaching infinity and so, can be advanced quickly. It should be noted that the integration time step size is, in practice, limited by the "goodness" of the solution desired. Information is lost as Δt is increased.

C.3 Systems of Coupled ODE's

Coupling in a set of ODE's is usually due to the physics of the situation under consideration. A second form of coupling, known as algorithmic coupling also sometimes exists. As this type of coupling is beyond the scope of this work, no further discussion will be given.

The solution techniques discussed to this point have assumed that the system is only mildly coupled. If the system happens to be more strongly coupled or "stiff", both the explicit and implicit methods have shortcomings.

Explicit

The explicit Euler procedure will require an extremely small integration time step to remain numerically stable. Nothing can be done about this. Moving to a more advanced explicit solution technique, such as a Runge-Kutta method, may allow the use of a larger time step, but an upper bound still exists. Clearly, computing costs are expected to be high when using any explicit method to solve a stiff system.

Implicit

Problems with the implicit procedure are noted with the matrix inversion. The more highly coupled the system of ODE's the more ill-conditioned (singular) the matrix inversion becomes. Fortunately, moving to a more advanced implicit algorithm usually avoids this problem.

Rearranging equation (C.13) into a new nonlinear function as follows:

$$f^*(u^{k+1})=u^{k+1}-u^k-\Delta t f^{k+1} \qquad (C.19)$$

Equation (C.19) is the residual error associated with a "guess" for u^{k+1}. Obviously, it is desirable that $f^*(u^{k+1})$ approach zero. This has reduced the problem to that of solving a nonlinear system of algebraic equations at each integration time step, avoiding a matrix inversion operation.

Several techniques for solving nonlinear systems of equations exist. Most are recursive in nature, and the Newton-Raphson method is considered to be the most "bullet proof". That is, it usually works best.

Newton-Raphson Iteration

The Newton-Raphson iteration is again derived from a truncated Taylor series expansion. In this case, the Taylor series is used to linearize the nonlinear function, $f^*(u^{k+1})$, as follows:

$$f^*(u^{k+1(m+1)})\approx f^*(u^{k+1(m)})+DuJ^{(m)} \qquad (C.20)$$

where:

$$Du=u^{k+1(m+1)}-u^{k+1(m)} \qquad (C.21)$$

$$J^{(m)}=d[f^*(u^{k+1(m)})]/du \qquad (C.22)$$

The bracketed superscript, m, is used to indicate the Newton-Raphson iteration counter. The recursive algorithm results from insisting that the left hand side of equation (C.20) be identically zero. That is, with rearrangement, (C.20) becomes:

$$DuJ^{(m)}=-f^*(u^{k+1(m)}) \qquad (C.23)$$

Solving this equation for Du, and then substitutine into equation (C.21) gives the updated value of $u^{k+1(m)}$:

$$u^{k+1(m+1)}=u^{k+1(m)}+Du \qquad (C.24)$$

As with nonlinear systems, there are many techniques available for solving linear systems of equations. Gauss-Jordan reduction is a simple, well known method and usually proves to be adequate. Details of this procedure are not given here as they may be found in any introductory numerical methods text book.

It should be clear now that the crux of the Newton-Raphson method is to iterate on equations (C.22), (C.23), and (C.24) until $u^{k+1(m+1)}=u^{k+1(m)}$ or until $f^*(u^{k+1(m)})$ is sufficiently close to zero.

Numerical Jacobian Evaluation

Although it is usually easiest and best to use an analytic version of the Jacobian, it is possible, and quite often necessary, to calculate its value numerically. The numerical technique involves estimating the Jacobian from difference quotients. That is, for column j of the Jacobian matrix, use:

$$\frac{df(u(t))}{du_j} = \frac{f(u(t)+u_{pj})-f(u(t))}{Du_j} \; ; \; j=1, \ldots, n \qquad (C.25)$$

where:

$$u_{pj}^T=[\ 0\ 0\ 0\ \ldots\ Du_j\ 0\ 0\ 0\ \ldots] \qquad (C.26)$$

These two equations amount to perturbing the dependent variable and observing the response. The approximation in equation (C.25) becomes truth as Du_j approaches zero. The perturbation vector, u_{pj}, is usually chosen such that $Du_j \simeq$ 1% of u_j itself. It should be noted that a lower limit must be placed on Du_j to prevent "divide by zero" if $u_j \simeq 0$.

Relaxation

If the system is especially stiff, a further modification to the Newton-Raphson technique may be required in order to converge on a solution to equation (C.24). This involves the inclusion of a weighting factor as follows:

$$u^{k+1(m+1)}=u^{k+1(m)}+wDu \qquad \text{(C.27)}$$

where:

$$w=[w_1, w_2, w_3, \ldots, w_n]; \; 0 \le w_i \le 2 \qquad \text{(C.28)}$$

w is normally termed a relaxation factor vector. Its purpose is to speed ($w_i > 1$) or slow ($w_i < 1$) the solution convergence to improve the numerical stability. In general, the choice of the various w_i will depend on the particular system being solved and a constant relaxation vector usually provides satisfactory results. However, w need not remain constant throughout the solution and more advanced methods exists for "controlling" w. As this is beyond the scope of this work, no further discussion will be given.

It is pertinent to put forth some words of caution to the reader at this point. Formost, is the fact that the function f^* must be differentiable; watch out for discontinuities. This becomes especially important if the Jacobian is being evaluated numerically. Also, the Jacobian matrix must be non-singular (i.e. det $J \neq 0$) if a solution is to exist. Relaxation is not a "fix all" technique. There are sets of equations that are too stiff even for this to work. Moving to "extended precision" computations may help. Finally, if the solution vector, u^{k+1}, is not unique, the answer obtained depends on the initial guess for u^{k+1}.

C.4 Translation of Theory into DYFLO/DYFLO2

Before the preceeding theory can be translated into the terminology Franks (1972, 1982) has used in his documentation, an alternate form for representing a set of ODE's must be considered. For most engineering applications, the ODE's of concern will be nonlinear and first order. By strict mathematical definition, the general form of such equations is:

$$u(t)\frac{du(t)}{dt} + b(t)u(t) = c(t) \qquad (C.29)$$

Rearranging this equation gives:

$$\frac{du(t)}{dt} = \frac{c(t)}{a(t)} - \frac{b(t)}{a(t)}u(t) \qquad (C.30)$$

This last equation is the form on which DYFLO and DYFLO2 have been based. That is, the "forcing function" is:

$$FF = \frac{c(t)}{a(t)} \qquad (C.31)$$

and the reciprocal time constant is:

$$DK = \frac{b(t)}{a(t)} \qquad (C.32)$$

so equation (C.28) may be written as:

$$\frac{du(t)}{dt} = FF - DK*u(t) \qquad (C.33)$$

It should be noted that a(t), b(t), and c(t) are quite often functions of u(t) which implies that FF and DK are functions of u(t) as well. As well, FF and DK are vectors in the multi-equation case. This is not clear in the documentation provided by Franks.

The right hasn side of equation (C.33) has been refered to as f(u(t)) in the previous discussion. In other words:

$$f(u(t)) = FF - DK*u(t) \qquad (C.34)$$

In view of equations (C.6) through (C.8), equation (C.34) can be rewritten as follows:

$$f^{k+1} = FF^{k+1} - DK^{k+1}*u^{k+1} \qquad (C.35)$$

Backshifting the time superscript and substituting equation (C.35) into equation (C.9) gives the recursive formula:

$$u^{k+1} = FF^{k}\Delta t - (1-DK^{k}\Delta t)*u^{k} \qquad (C.36)$$

This equation corresponds directly to equation I-5 in Franks documentation (1982) if it is realized that "n" is used as the time step counter and Dt is actually Δt. The DYFLO routine, INTG, is based on equation (C.36).

In a similar fashion, equation (C.35) can be substituted into the recursive formula for the implicit Euler method (equation (C.13)):

$$u^{k+1} = u^k + \Delta t(FF^{k+1} - DK^{k+1} * u^{k+1}) \qquad (C.37)$$

Solving (C.37) for u^{k+1} yields:

$$u^{k+1} = (u^k + \Delta t FF^{k+1})/(1+\Delta t DK^{k+1}) \qquad (C.38)$$

which corresponds to Franks' (1982) equation I-9. This is the formula on which the DYFLO2 routine INTSTF is based. It quickly becomes obvious that there are some problems with equation (C.38). Advance knowledge of both FF^{k+1} and DK^{k+1} is required, and can not be calculated if they are functions of u(t), as mentioned earlier. Franks suggests that if FF and DK are "mostly invariable", FF^k and DK^k may be used, which should prove to be adequate for a single ODE. However, if FF and/or DK are even mildly time varying or a set of equations is involved, a predictor corrector solution procedure is outlined by Franks. The implicit formula is used with data available at time step k to "predict" the solution at k+1. The predicted data is the used in the implicit formula again to "correct" the solution. What Franks has failed to mention is the fact that by using the implicit formula in a predictor role, one is really using the explicit formula and time step limitations have been reintroduced in the process. A better approach, which avoids this problem, is given in the section on highly coupled systems (Franks, 1982). The procedure outlined is the same as that presented earlier in this work for stiff systems.

Substituting equation (C.35) into equation (C.18) (the residual error function) gives:

$$f^{*}(u^{k+1})=u^{k+1}-u^{k}-\Delta t(FF^{k+1}-DK^{k+1}*u^{k+1}) \qquad (C.39)$$

Multiplying this equation out, and then rearranging it gives:

$$f^{*}(u^{k+1})=u^{k+1}(I+\Delta tDK^{k+1})-u^{k}+\Delta tFF^{k+1}) \qquad (C.40)$$

Equation (C.40) uses a rigorous vector notation and corresponds to equation (V-5) in Franks (1982). Equation (V-6) as given by Franks (1982) corresponds to equation (C.23) for the case of three ODE's and the Newton-Raphson iteration equation is not numbered, but is identical to the one given in this work (cf. equation (C.24)).

Relaxation in the Newton-Raphson iteration is only briefly discussed by Franks. He calls the relaxation factor vector an attenuation factor and equation (V-7) in his work corresponds to equation (C.27). An important point that has been left out of the documentation is that is is generally accepted that the relaxation factor must be bounded (i.e. $0 \le w \le 2$) for numerical stability.

The final point to be noted is that Franks indicates that the Jacobian is required in an anlytic form. As was show earlier in this work, the Jacobian matrix can be reliably estimated numerically.

C.5 Recursive Integration Algorithm Summary

The complete recursive integration algorithm suggested by Franks (1982) is given in detail below:

i. Make an initial guess for the solution, u^{k+1}. The implicit formula at time step k is used in a predictor sense to automate this guessing procedure.

ii. Compute the residual error function from equation (C.40). If f^* is sufficiently close to zero, go to step 6.

iii. Calculate the Jacobian matrix via equation (C.22). Note, this is done using the most "up to date" value of u^{k+1} available.

iv. Solve the system of equations in (C.23) for Du. Relax this increment if necessary before updating u^{k+1} in equation (C.24).

v. Repeat steps 2 through 4.

vi. Proceed to the next time step calculation. (i.e. $k=k+1 \longrightarrow k+1$).

Appendix D: New DYFLO2.LIB Subroutine Details

D.1 SUBROUTINE CON2X2

Purpose: Continuous 2x2 controller subroutine.

Use: CALL CON2X2(CO1,CO2,CI1,CI2,MODE1,MODE2)

Parameters: See source code listing in Figure D.1.

Additional Input Data Required from Labelled COMMON:

COMMON/LOOP1/
SP1 - loop 1 setpoint, same units as controller output input signal, CI1.

COMMON/LOOP2/
SP2 - loop 2 setpoint same units as controller output input signal, CI2.

COMMON/CO2BY2/
ZR1 - loop 1 zero input scale.
ZR2 - loop 2 zero input scale.
FSRDG1 - loop 1 full input scale.
FSRDG2 - loop 2 full input scale.
AXN1 - loop 1 controller action switch.
AXN2 - loop 2 controller action switch.
PB1 - loop 1 proportional band.
PB2 - loop 2 proportional band.
RPT1 - loop 1 repeat rate (= 0 if MODE1 = 1).
RPT2 - loop 2 repeat rate (= 0 if MODE2 = 1).
RT1 - loop 1 rate time (derivative time constant). (= 0 if MODE1 = 1 or 2)
RT2 - loop 2 rate time (derivative time constant). (= 0 if MODE1 = 1 or 2)
RA1 - loop 1 rate amplitude (= 0 if MODE1 = 1 or 2).
RA2 - loop 2 rate amplitude (= 0 if MODE1 = 1 or 2).
OI1 - loop 1 integral action or controller bias, initialize to steady state controller output.
OI2 - loop 2 integral action or controller bias, initialize to steady state controller output.
ODI1 - loop 1 derivative action output, initialize to 1./RA1 - 1. if MODE1 = 3.
ODI2 - loop 2 derivative action output, initialize to 1./RA2 - 1. if MODE2 = 3.

Brief Description:

The subroutine, CON2X2, provides a convenient method of implimenting a continuous 2x2 control scheme. Proportional, proportional plus integral, and proportional, integral, derivative controller modes can be used with either control loop by setting the flags, MODE1 and MODE2, to 1, 2, or 3 (see Figure D.1). The subroutine calls the appropriate DYFLO2 controller subroutine (CONTR1, CONTR2, or CONTR3) to perform the controller calculations for the particular loop. Figure D.1 shows a source code listing of the subroutine and Figure D.2 presents a program flow chart which will clarify the logic used.

D.2 SUBROUTINE DISPID

Purpose: Discrete PID controller using the velocity formulation with derivative action only on the input signal.

Use: CALL DISPID(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,
TIM,N,TO)

Parameters: See source code listing in Figure D.3.

Additional Input Data Required from Labelled COMMON:

None.

Brief Description:

The subroutine, DISPID, provides discrete control for up to 25 loops, on a single loop basis. Proportional, proportional plus integral, and proportional, integral , derivative controller modes can be used by setting the MODE flag appropriately. The structure of the controller is given in Figure 2.8 and the calculations performed by this subroutine are based on equation (2.22). The program flow chart in Figure D.4 will clarify the logic used in this

```
C ********* CON2X2 *******************************************
C
C   CONTINUOUS 2 BY 2 CONTROLLER ROUTINE
C
C-------------------------------------------------------------
C
C   VARIABLE DESCRIPTION
C
C   INPUT
C   CI1    - LOOP 1 CONTROLLER INPUT
C   CI2    - LOOP 2 CONTROLLER INPUT
C   MODE1  - LOOP 1 CONTROLLER MODE   1---> P
C                                     2---> PI
C                                     3---> PID
C   MODE2  - LOOP 2 CONTROLLER MODE   1---> P
C                                     2---> PI
C                                     3---> PID
C
C   OUTPUT
C   CO1    - LOOP 1 CONTROLLER OUTPUT (0 - 100%)
C   CO2    - LOOP 2 CONTROLLER OUTPUT (0 - 100%)
C
C-------------------------------------------------------------
C
      SUBROUTINE CON2X2(CO1,CO2,CI1,CI2,MODE1,MODE2)
      REAL KC1,KC2
      COMMON/LOOP1/ KC1,TI1,TD1,SP1,CSP1,DIST1,TS1
      COMMON/LOOP2/ KC2,TI2,TD2,SP2,CSP2,DIST2,TS2
      COMMON/CO2BY2/ ZR1,ZR2,FSRDG1,FSRDG2,AXN1,AXN2,PB1,PB2,
     + RPT1,RPT2,RT1,RT2,RA1,RA2,OI1,OI2,ODI1,ODI2
C
C   LOOP 1 CONTROLLER CALCULATIONS
C
      GOTO(10,20,30),MODE1
   10 CALL CONTR1(CI1,CO1,ZR1,FSRDG1,SP1,AXN1,PB1,OI1)
      GOTO 40
   20 CALL CONTR2(CI1,CO1,ZR1,FSRDG1,SP1,AXN1,PB1,RPT1,OI1)
      GOTO 40
   30 CALL CONTR3(CI1,CO1,ZR1,FSRDG1,SP1,AXN1,PB1,RPT1,RT1,RA1,
     +            OI1,ODI1)
C
C   LOOP 2 CONTROLLER CALCULATIONS
C
   40 GOTO(50,60,70),MODE2
   50 CALL CONTR1(CI2,CO2,ZR2,FSRDG2,SP2,AXN2,PB2,OI2)
      GOTO 80
   60 CALL CONTR2(CI2,CO2,ZR2,FSRDG2,SP2,AXN2,PB2,RPT2,OI2)
      GOTO 80
   70 CALL CONTR3(CI2,CO2,ZR2,FSRDG2,SP2,AXN2,PB2,RPT2,RT2,RA2,
     +            OI2,ODI2)
   80 RETURN
      END
```

Figure D.1: Subroutine CON2X2 FORTRAN IV Source Code Listing

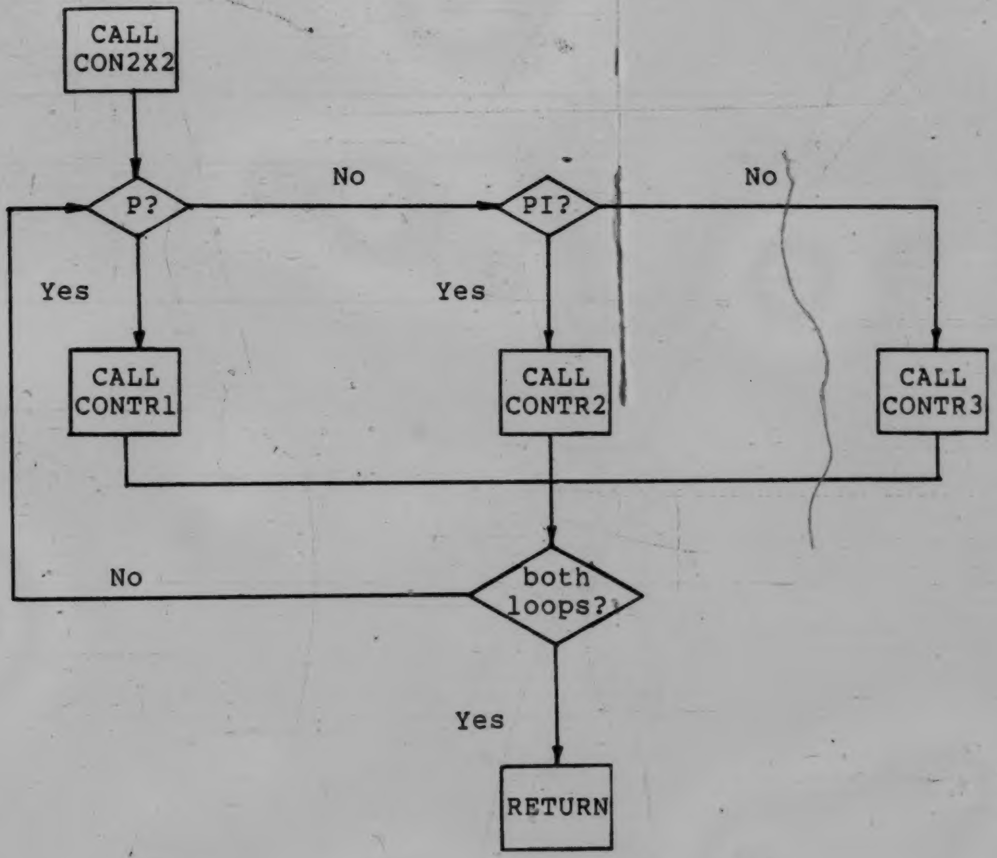

CALL CON2X2
P?
No
PI?
No
Yes
Yes
CALL CONTR1
CALL CONTR2
CALL CONTR3
No
both loops?
Yes
RETURN

Figure D.2: Subroutine CON2X2 Program Flow Chart

```
C*********** DISPID *********************************************
C
C   DISCRETE PID CONTROLLER ROUTINE
C
C   NOTE: VELOCITY FORM, DERIVATIVE ON OUTPUT STRUCTURE
C-----------------------------------------------------------------
C
C   PARAMETERS
C
C    VI     - process input signal
C    CO     - controller output signal, incremental
C    ZR     - process input zero reading
C    FSRDG  - process input full scale reading
C    SP     - setpoint, incremental
C    AXN    - comparator action  +1---> direct action
C                                -1---> inverse action
C    KP     - proportional gain
C    KI     - integral gain
C    KD     - derivative gain
C    TS     - sample time
C    MODE   - controller mode   1---> P
C                               2---> P+I
C                               3---> P+I+D
C    TIN    - simulation time
C    N      - discrete PID controller number
C   TO      - sample time offset
C
C   INTERNAL
C
C    STIN   - simulation time for next sample
C    NS     - sample counter
C    TEST   - take new sample flag
C    SPAN   - input variable span
C    VIN    - dummy input variable
C    PC     - proportional control output
C    IC     - integral control output
C    ER     - current sample error
C    ERN1   - 1 sample previous error
C    PCIN   - input signal in percent
C    PCINN1 - 1 sample previous input signal in percent
C    PCINN2 - 2 samples previous input signal in percent
C
C-----------------------------------------------------------------
C
      SUBROUTINE DISPID(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,TIN,N,TO)
      LOGICAL IFLAG(25)/25*.TRUE./
      REAL PCIN(25),PCINN1(25),PCINN2(25),ER(25),ERN1(25),IC,KP,KI,KD
      REAL ERN2(25)
      INTEGER NS(25)/25*0/
      DATA ER/25*0./,ERN1/25*0./,PCINN1/25*0./,PCINN2/25*0./,
     + PCIN/25*0./,ERN2/25*0./
      STIN=NS(N)*TS+TO
      TEST=TIN-STIN
C      IF(ABS(TEST) .LT. 5E-5) TEST=0.0
      IF(TEST) 40,5,5
    5 SPAN=ABS(FSRDG-ZR)
      VIN=VI
      IF(VI .GT. FSRDG) VIN=FSRDG
      IF(VI .LT. ZR) VIN=ZR
      IF(IFLAG(N)) PCINN2(N)=VIN/SPAN*(-1.)*AXN
      IF(IFLAG(N)) PCINN1(N)=VIN/SPAN*(-1.)*AXN
      IF(IFLAG(N)) PCIN(N)=VIN/SPAN*(-1.)*AXN
      IC=0.
      PC=0.
      GOTO(20,20,10),MODE
   10 PCINN2(N)=PCINN1(N)
      PCINN1(N)=PCIN(N)
      PCIN(N)=VIN/SPAN*(-1.)*AXN
      DC=KD*(-PCIN(N)+2.*PCINN1(N)-PCINN2(N))/TS
      IF(ABS(DC) .LT. .001) DC=0.0
      VIN=DC+VIN
   20 ERN1(N)=ER(N)
      ER(N)=(SP-VIN)/SPAN*(-100.)*AXN
      IF(MODE .EQ. 1) GOTO 30
      IC=KI*TS*(ER(N)+ERN1(N))/2.
   30 PC=KP*(ER(N)-ERN1(N))
      CO=CO+PC+IC
      IF(CO .GE. 100.) CO=100.
      IF(CO .LT. 0.) CO=0.
      NS(N)=NS(N)+1
      IFLAG(N)=.FALSE.
   40 RETURN
      END
```

Figure D.3: Subroutine DISPID FORTRAN IV Source Code Listing

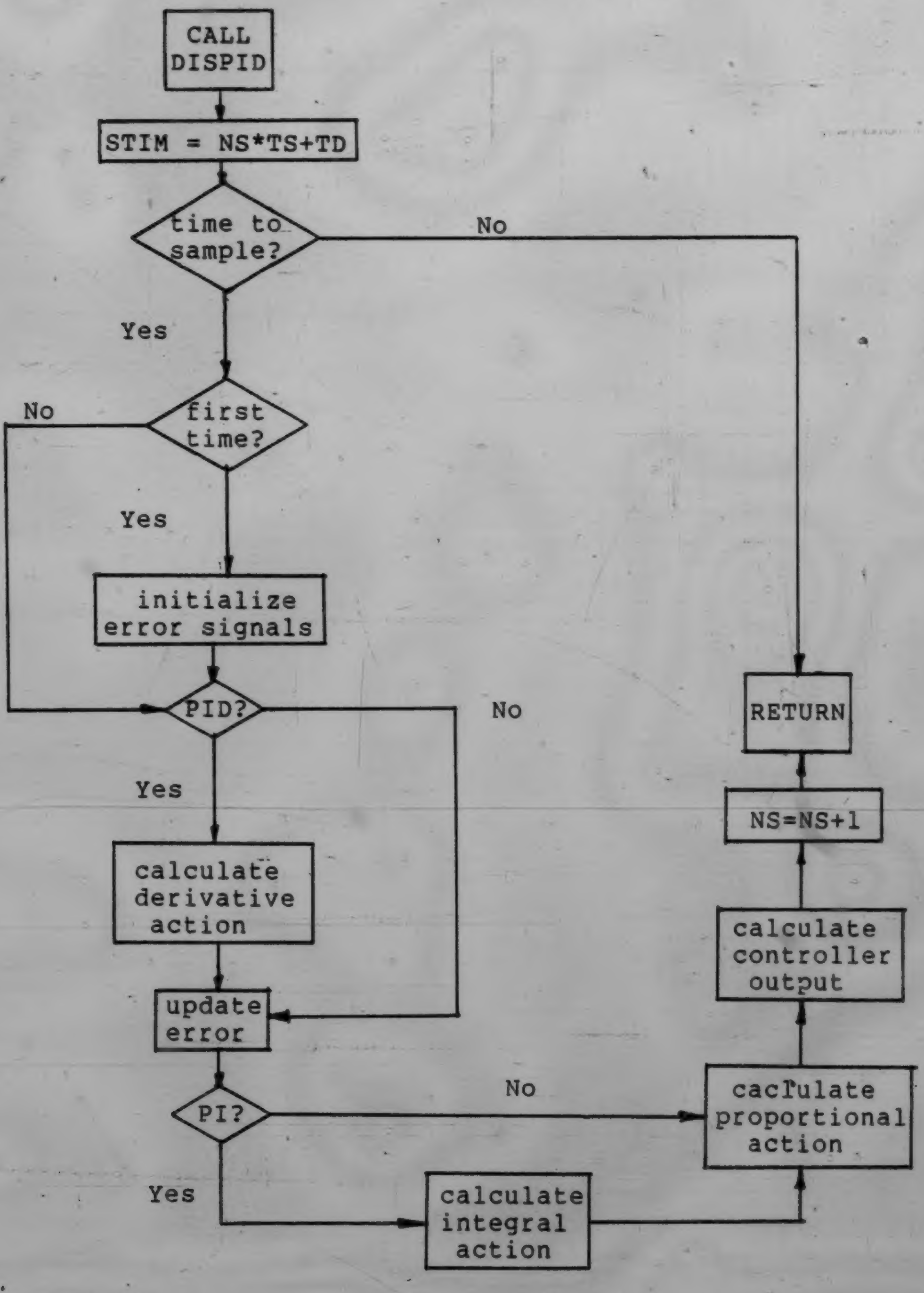

CALL DISPID
STIM = NS*TS+TD
time to sample?
No
Yes
first time?
No
Yes
initialize error signals
PID?
No
Yes
calculate derivative action
update error
PI?
No
Yes
calculate integral action
caclulate proportional action
calculate controller output
NS=NS+1
RETURN

Figure D.4: Subroutine DISPID Program Flow Chart

subroutine.

D.3 SUBROUTINE DPID2

Purpose: Discrete PID controller using the velocity formulation with the standard textbook structure.

Use: CALL DPID2(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,
TIM,N)

Parameters: See source code listing in Figure D.5.

Additional Input Date Required from Labelled COMMON:

None.

Brief Description:

The subroutine, DPID2, provides discrete control for up to 25 loops, on a single loop basis. The program logic is similar to DISPID (cf. Figure D.4). The structure of the controller is given in Figure 2.7 and the calculations performed by this subroutine are based on equation (2.21).

D.4 SUBROUTINE I2BY2

Purpose: An interacting 2x2 first order transfer function model.

Use: CALL I2BY2(TIM,RINT,N)

Parameters: See source code listing in Figure D.6.

Additional Input Data Required from Labelled COMMON:

COMMON/IN2BY2/
KP - process transfer function gain matrix (2x2)
KL - load transfer function gain vector (2)
TAU - process time constant matrix (2x2)
TAUL - load time constant vector (2)
NTL - process time delay as an integer number of integration time intervals (2x2)
NTLL - load time delay as an integer number of integration time intervals (2x2)
DX - process deviation input vector (2)
DY - process deviation output vector (2)
DD - disturbance deviation input vector (2)

```
C ********* DPID2 ***********************************************
C
C    DISCRETE PID CONTROLLER ROUTINE
C
C    "TEXTBOOK" STRUCTURE
C---------------------------------------------------------------
C
C    PARAMETERS
C
C     VI    - process input signal
C     CO    - controller output signal, incremental
C     ZR    - process input zero reading
C     FSRDG - process input full scale reading
C     SP    - setpoint, incremental
C     AXN   - comparator action  +1---> direct action
C                                -1---> inverse action
C     KP    - proportional gain
C     KI    - integral gain
C     KD    - derivative gain
C     TS    - sample time
C     MODE  - controller mode   1---> P
C                               2---> P+I
C                               3---> P+I+D
C     TIM   - simulation time
C     N     - discrete PID controller number
C
C    INTERNAL
C
C     STIM  - simulation time for next sample
C     NS    - sample counter
C     TEST  - take new sample flag
C     SPAN  - Input variable span
C     VIN   - dummy input variable
C     PC    - proportional control output
C     IC    - integral control output
C     DC    - derivative control output
C     ER    - current sample error
C     ERM1  - 1 sample previous error
C     PCIN  - input signal in percent
C     PCINM1 - 1 sample previous input signal in percent
C     PCINM2 - 2 samples previous input signal in percent
C
C---------------------------------------------------------------
C
      SUBROUTINE DPID2(VI,CO,ZR,FSRDG,SP,AXN,KP,KI,KD,TS,MODE,TIM,N)
      REAL PCIN(25),PCINM1(25),PCINM2(25),ER(25),ERM1(25),IC,KP,KI,KD
      REAL ERM2(25)
      INTEGER NS(25)/25*0/
      DATA ER/25*0./,ERM1/25*0./,PCINM1/25*0./,PCINM2/25*0./,
     + PCIN/25*0./,ERM2/25*0./
      IF(TIM .LE. 0.) GOTO 5
      STIM=NS(N)*TS
      TEST=TIM-STIM
      IF(TEST) 40,5,5
    5 SPAN=ABS(FSRDG-ZR)
      VIN=VI
      IF(VI .GT. FSRDG) VIN=FSRDG
      IF(VI .LT. ZR) VIN=ZR
      DC=0.
      IC=0.
      PC=0.
      PCINM2(N)=PCINM1(N)
      PCINM1(N)=PCIN(N)
      PCIN(N)=VIN/SPAN*(-100.)*AXN
      ERM2(N)=ERM1(N)
      ERM1(N)=ER(N)
      ER(N)=(SP-VIN)/SPAN*(-100.)*AXN
      GOTO(30,20,10),MODE
   10 DC=KD*(ER(N)-2.*ERM1(N)+ERM2(N))/TS
   20 IC=KI*TS*(ER(N)+ERM1(N))/2.
   30 PC=KP*(ER(N)-ERM1(N))
      CO=CO+PC+IC+DC
      IF(CO .GE. 100.) CO=100.
      IF(CO .LT. 0.) CO=0.
      NS(N)=NS(N)+1
   40 RETURN
      END
```

Figure D-5: Subroutine DPID2 FORTRAN IV Source Code Listing

Brief Description

The subroutine, I2BY2, provides a convenient method for modelling a 2x2 interacting process with first order plus time delay transfer functions. The subroutine accomplishes this with six calls to the DYFLO2 subroutines TFN1 and TDL in the appropriate order. Figure D.6 shows a source code listing of the subroutine and Figure D.7 presents a program flow chart which will clarify the logic used.

D.5 SUBROUTINE LLCOMP

Purpose: Lead-lag compensation avoiding the use of the numerical derivative.

Use: CALL LLCOMP(YOUT,XIN,TC1,TC2,TD,GAIN,NO)

Parameters: See source code listing in Figure D.8.

Additional Input Data Required from Labelled COMMON:

COMMON/CINT/
DT - integration time interval

Brief Description:

The subroutine, LLCOMP, provides up to 25 lead lag units, including time delay and gain elements, for use in control simulation programs. The algorithm used in this implementation follows equations (2.38) through (2.40) where the numerical derivative calculations are avoided. This subroutine makes use of the DYFLO2 routines INTG, INTSTF, and TDL and the program flow chart in Figure D.9 will clarify the logic used in this subroutine.

D.6 SUBROUTINE LLCDER

```
C ***** I2BY2 ***************************************************
C
C   INTERACTING TWO BY TWO FIRST ORDER TRANSFER FUNCTION
C   MODEL ROUTINE
C
C---------------------------------------------------------------
C
C   PARAMETERS
C
C   TIN  - CURRENT SIMULATION TIME
C   RINT - INTEGRATION INTERVAL
C   N    - ROUTINE CALL NUMBER
C   KP   - PROCESS GAIN MATRIX
C   KL   - LOAD GAIN VECTOR
C   TAU  - PROCESS TIME CONSTANT MATRIX
C   TAUL - LOAD TIME CONSTANT VECTOR
C   NTL  - PROCESS TIME DELAY IN NUMBER OF INTEGRATION INTERVALS
C   NTLL - LOAD TIME DELAY IN NUMBER OF INTEGRATION INTERVALS
C   NTLC - TIME DELAY LOCATION COUNTER
C   DX   - INPUT VECTOR (DEVIATION VARIABLES)
C   DY   - OUTPUT VECTOR (DEVIATION VARIABLES)
C   DD   - LOAD VECTOR (DEVIATION VARIABLES)
C   YA   - TIME DELAY WORK AREA
C   IFLAG- INITIAL CALL FLAG
C--------------------------------------------------------------
C
      SUBROUTINE I2BY2(TIN,RINT,N)
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      REAL KP(2,2),KL(2),TAU(2,2),TAUL(2),DX(2),DY(2),DD(2),YA(25,500)
      REAL L1,L2
      INTEGER NTL(2,2),NTLL(2),NTLC(25)/25*0/
      LOGICAL IFLAG(25)/25*.FALSE./
      COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
C
C   TEST FOR FIRST CALL
C
      IF(IFLAG(N)) GOTO 10
      Y11=0.
      Y12=0.
      Y21=0.
      Y22=0.
      L1=0.
      L2=0.
      IFLAG(N)=.TRUE.
C
C   G11
C
   10 CALL TFN1(Y11,DX(1),TAU(1,1),KP(1,1))
      CALL TDL(YA,NTLC,Y11,DY11,NTL(1,1),1)
      IF(NTL(1,1) .EQ. 0) DY11=Y11
C
C   G12
C
      CALL TFN1(Y12,DX(2),TAU(1,2),KP(1,2))
      CALL TDL(YA,NTLC,Y12,DY12,NTL(1,2),2)
      IF(NTL(1,2) .EQ. 0) DY12=Y12
C
C   G21
C
      CALL TFN1(Y21,DX(1),TAU(2,1),KP(2,1))
      CALL TDL(YA,NTLC,Y21,DY21,NTL(2,1),3)
      IF(NTL(2,1) .EQ. 0) DY21=Y21
C
C   G22
C
      CALL TFN1(Y22,DX(2),TAU(2,2),KP(2,2))
      CALL TDL(YA,NTLC,Y22,DY22,NTL(2,2),4)
      IF(NTL(2,2) .EQ. 0) DY22=Y22
C
C   GL1
C
      CALL TFN1(L1,DD(1),TAUL(1),KL(1))
      CALL TDL(YA,NTLC,L1,DL1,NTLL(1),5)
      IF(NTLL(1) .EQ. 0) DL1=L1
C
C   GL2
C
      CALL TFN1(L2,DD(2),TAUL(2),KL(2))
      CALL TDL(YA,NTLC,L2,DL2,NTLL(2),6)
      IF(NTLL(2) .EQ. 0) DL2=L2
C
C   CALCULATE FINAL OUTPUT
C
      DY(1)=DY11+DY12+DL1
      IF(ABS(DY(1)) .LT. 1.E-10) DY(1)=0.
      DY(2)=DY22+DY21+DL2
      IF(ABS(DY(2)) .LT. 1.E-10) DY(2)=0.
      RETURN
      END
```

Figure D.6: Subroutine I2BY2 FORTRAN IV Source Code Listing

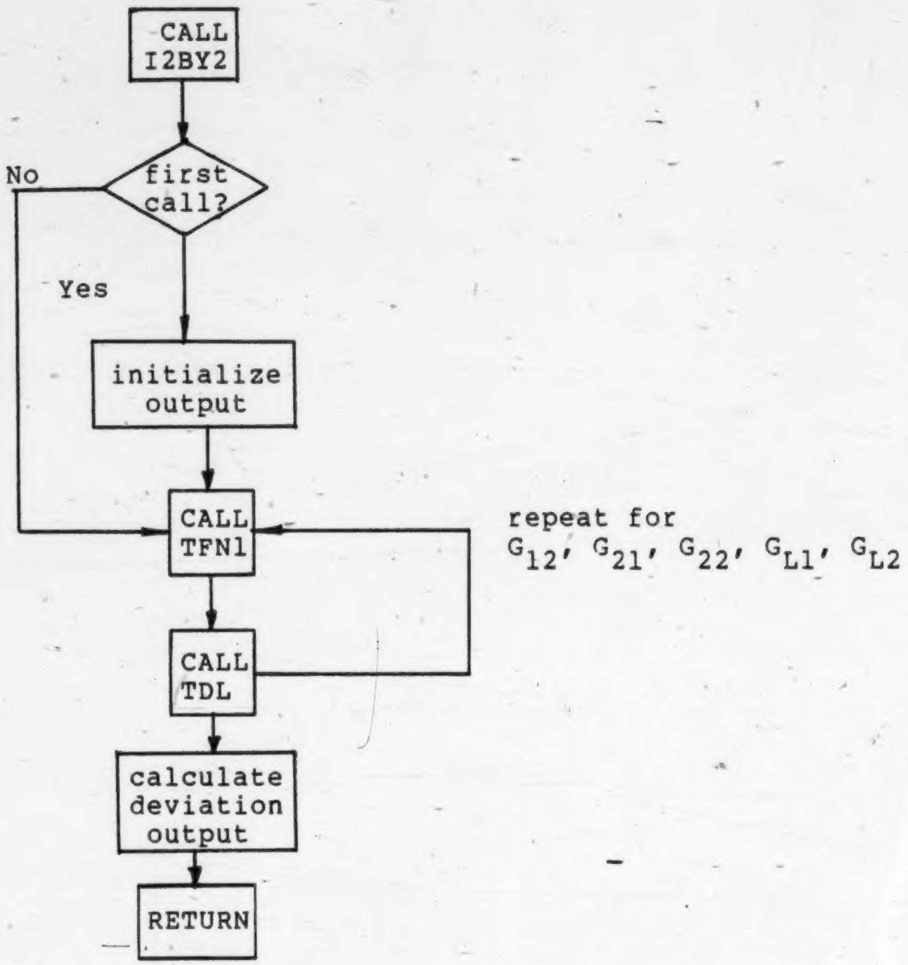

CALL I2BY2
first call?
No
Yes
initialize output
CALL TFN1
CALL TDL
repeat for G_{12}, G_{21}, G_{22}, G_{L1}, G_{L2}
calculate deviation output
RETURN

Figure D.7: Subroutine I2BY2 Program Flow Chart

```
C ***** LLCOMP ************************************************
C
C   LEAD-LAG COMPENSATOR ROUTINE
C
C   REFERENCE: SMITH & CORRIPIO; PRINCIPLES AND PRACTICE OF
C              AUTOMATIC PROCESS CONTROL, JOHN WILEY & SONS,
C              1985, pp. 497-498.
C
C--------------------------------------------------------------
C
C   PARAMETERS
C
C   YOUT  - OUTPUT SIGNAL
C   XIN   - INPUT SIGNAL
C   TC1   - LEAD TIME CONSTANT
C   TC2   - LAG TIME CONSTANT
C   TD    - TIME DELAY
C   GAIN  - COMPENSATOR GAIN
C   NO    - LEAD-LAG NUMBER (MAX. OF 25)
C
C   INTERNAL
C
C   ICF   - INITIAL CONDITION SET FLAG
C   Y     - UNDELAYED OUTPUT SIGNAL (STATE)
C   Y1    - INTERMEDIATE OUTPUT SIGNAL (STATE)
C   DY1   - DERIVATIVE OF INTERMEDIATE OUTPUT SIGNAL
C   DX    - DERIVATIVE OF INPUT SIGNAL
C   FF    - FORCING FUNCTION
C   TK    - RECIPROCAL OF THE TIME CONSTANT
C   NTL   - NUMBER OF INTEGRATION INTERVALS IN DELAY
C   NTLC  - TIME DELAY LOCATION COUNTER
C   YA    - TIME DELAY WORK AREA
C   DT    - INTEGRATION INTERVAL
C   YY    - DUMMY INPUT FOR TIME DELAY INITIALIZATION
C
C--------------------------------------------------------------
C
      SUBROUTINE LLCOMP(YOUT,XIN,TC1,TC2,TD,GAIN,NO)
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      REAL YA(25,500),Y(25),Y1(25)/25*0./
      LOGICAL ICF(25)/25*.FALSE./
      INTEGER NTLC(25)/25*0/
      IF(ICF(NO)) GOTO 5
      IF(TD .LE. 0.) GOTO 4
      YY=YOUT
      NTL=INT(TD/DT)
      CALL TDL(YA,NTLC,YY,YOUT,NTL,NO)
    4 ICF(NO)=.TRUE.
      Y1(NO)=YOUT
    5 Y(NO)=Y1(NO)+TC1*XIN/TC2
      IF(DT .GT. TC1 .OR. DT .GT. TC2) GOTO 10
      DY1=(XIN-Y(NO))/TC2
      CALL INTG(Y1(NO),DY1)
      GOTO 20
   10 FF=XIN/TC2
      TK=1./TC2
      CALL INTSTF(Y1(NO),TK,FF)
   20 IF(TD .LE. 0.) GOTO 30
      NTL=INT(TD/DT)
      CALL TDL(YA,NTLC,Y(NO),YOUT,NTL,NO)
      YOUT=YOUT*GAIN
      GOTO 40
   30 YOUT=Y(NO)*GAIN
   40 RETURN
      END
```

Figure D.8: Subroutine LLCOMP FORTRAN IV Source Code Listing

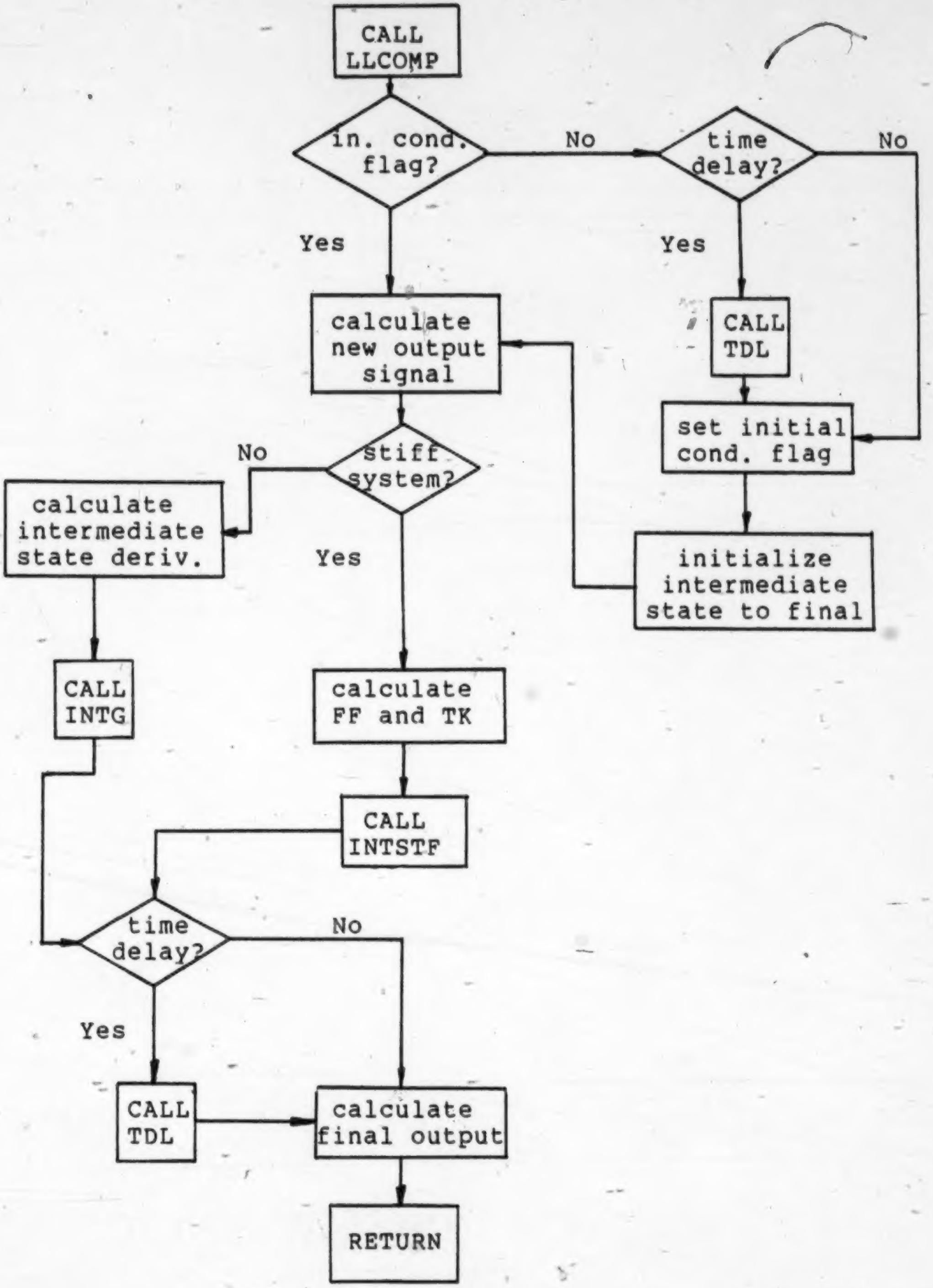

CALL LLCOMP
in. cond. flag?
No
Yes
time delay?
No
Yes
calculate new output signal
CALL TDL
set initial cond. flag
No
stiff system?
calculate intermediate state deriv.
Yes
initialize intermediate state to final
CALL INTG
calculate FF and TK
CALL INTSTF
time delay?
No
Yes
CALL TDL
calculate final output
RETURN

Figure D.9 Subroutine LLCOMP Program Flow Chart

Purpose: Lead-lag compensation using the numerical derivative implementation.

Use: CALL LLCDER(YOUT,XIN,TC1,TC2,TD,GAIN,NO,TNO,RINT)

Parameters: See source code listing in Figure D.10.

Additional Input Data Required from Labelled COMMON:

None.

Brief Description:

The subroutine, LLCDER, provides up to 25 lead lag units, including time delay and gain elements, for use in control simulation programs. The algorithm used for this implementation involves the derivative of the input signal and is given in equation (2.32). The program logic is similar to that used in LLCOMP, with the exception that the DYFLO2 routine DER is used to numerically calculate the derivative mentioned above.

D.7 SUBROUTINE PTRAIN

Purpose: Generate a square wave pulse train.

Use: CALL PTRAIN(OUT,P,PTIM,TIM,NO)

Parameters: See source code listing in Figure D.11.

Additional Input Data Required from Labelled COMMON:

None.

Brief Description:

The subroutine, PTRAIN, provides a convenient method for defining a pulse train input for use in simulating control strategy behavior. The normalized pulse definition vector, P, consists of a combination of fifty -1's, 0's, and +1's,

```
C *********  LLCDER  ****************************************
C
C   LEAD-LAG COMPENSATOR ROUTINE
C
C
      SUBROUTINE LLCDER(YOUT,XIN,TC1,TC2,TD,GAIN,TIM,NO,TNO,RINT)
C
C------------------------------------------------------------
C
C   PARAMETERS
C
C   YOUT  - OUTPUT SIGNAL
C   XIN   - INPUT SIGNAL
C   TC1   - LEAD TIME CONSTANT
C   TC2   - LAG TIME CONSTANT
C   TD    - DEAD TIME
C   GAIN  - COMPENSATOR GAIN
C   TIM   - SIMULATION TIME
C   NO    - LEAD-LAG NUMBER - MAX. 10
C   TNO   - TIME DELAY NUMBER
C   RINT  - INTEGRATION INTERVAL
C
C   INTERNAL
C
C   Y     - UNDELAYED OUTPUT SIGNAL
C   DY    - DERIVATIVE OF OUTPUT SIGNAL
C   DX    - DERIVATIVE OF INPUT SIGNAL
C   FF    - FORCING FUNCTION
C   TK    - INVERSE TIME CONSTANT
C   NTL   - NUMBER OF INTEGRATION INTERVALS IN DELAY
C
C------------------------------------------------------------
C
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      INTEGER TNO
      REAL YA(25,500),Y(25)
      IF(TIM .LE. 0.) Y(NO)=YOUT
      CALL DER(XIN,TIM,DX,NO)
      IF(DT .GT. TC1 .OR. DT .GT. TC2) GOTO 10
      DY=(GAIN*(DX*TC1+XIN)-Y(NO))/TC2
      CALL INTG(Y(NO),DY)
      GOTO 20
   10 FF=GAIN*(DX*TC1+XIN)/TC2
      TK=1./TC2
      CALL INTSTF(Y(NO),TK,FF)
   20 IF(TD .LE. 0.) GOTO 30
      NTL=INT(TD/RINT)
      CALL TDL(YA,Y(NO),YOUT,NTL,TNO)
      GOTO 40
   30 YOUT=Y(NO)
   40 RETURN
      END
```

Figure D.10: Subroutine LLCDER FORTRAN IV Source Code Listing

```
C ********* PTRAIN **********************************************
C
C   SQUARE WAVE PULSE TRAIN GENERATION ROUTINE
C
C---------------------------------------------------------------
C
C   PARAMETERS
C
C   OUT     - output signal
C   AMPL    - amplitude of pulse wave
C   P       - normalized pulse definition vector
C   PTIM    - pulse period
C   TIM     - current simulation time
C   NO      - routine call index
C
C   INTERNAL
C   N       - sample interval counter
C   PTIMN   - simulation time for next sample
C   TEST    - change pulse signal flag
C
C---------------------------------------------------------------
C
      SUBROUTINE PTRAIN(OUT,AMPL,P,PTIM,TIM,NO)
      REAL P(50)
      INTEGER N(25)/25*0/
      IF(TIM .LE. 0.) GOTO 20
      PTIMN=N(NO)*PTIM
      TEST=TIM-PTIMN
      IF(TEST) 10,20,20
   10 RETURN
   20 N(NO)=N(NO)+1
      OUT=P(N(NO))*AMPL
      RETURN
      END
```

Figure D.11: Subroutine PTRAIN FORTRAN IV Source Code Listing

ordered to produce the desired pulse train shape. The amplitude of the pulse is the absolute value of the desired magnitude, while the period, PTIM, is a constant in units consistent with the total simulation time, TIM. Up to 25 different pulse trains may be used in a given simulation program. The program flow chart presented in Figure D.12 will clarify the logic used in this subroutine.

D.8 SUBROUTINE RATCON

Purpose: Ration controller.

Use: CALL RATCON(COR,CIR,RATIO)

Parameters: See source code listing in Figure D.13.

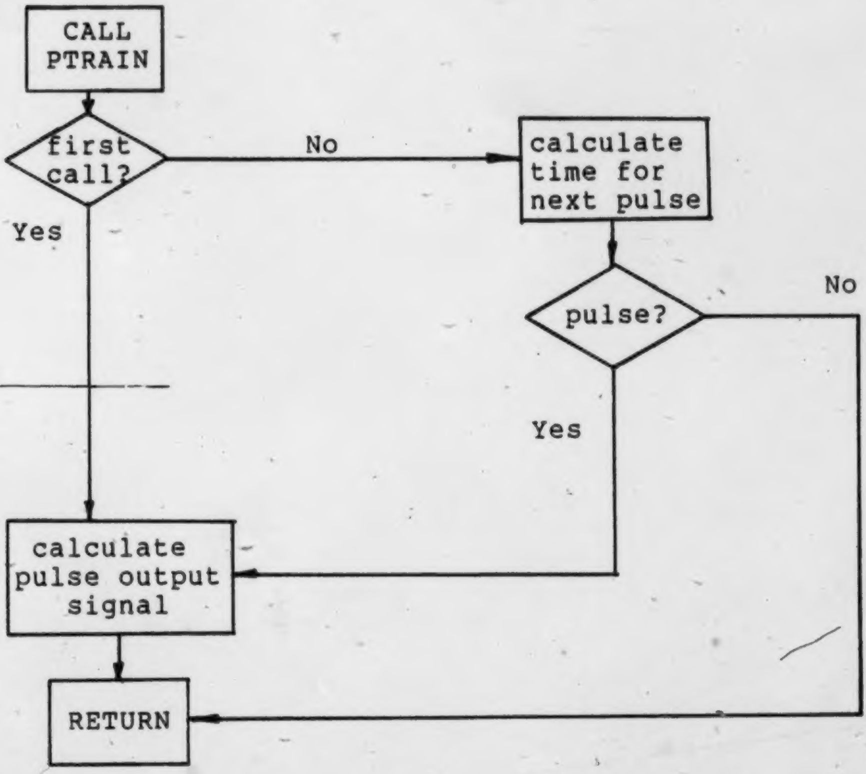

CALL PTRAIN
first call?
No
Yes
calculate time for next pulse
pulse?
No
Yes
calculate pulse output signal
RETURN

Figure D.12: Subroutine PTRAIN Program Flow Chart

```
C ******** RATCON *********************************************
C
C   RATIO CONTROLLER ROUTINE
C
C--------------------------------------------------------------
C
C   VARIABLE DESCRIPTION
C
C   INPUT
C   CIR    - INPUT TO RATIO CONTROLLER
C   RATIO  - RATIO INPUT TO OUTPUT (CIR/COR)
C
C   OUTPUT
C   COR    - OUTPUT FROM RATIO CONTROLLER
C
C--------------------------------------------------------------
C
      SUBROUTINE RATCON(COR,CIR,RATIO)
      COR=CIR/RATIO
      RETURN
      END
```

Figure D.13: Subroutine RATCON FORTRAN IV Source Code Listing

Additional Input Data Required from Labelled COMMON:

None.

Brief Description:

The subroutine, RATCON, provides a ratio controller for use in DYFLO2 simulation programs. The output of the subroutine is simply the input signal divided by the ratio factor. The program flow chart in Figure D.14 will clarify the subroutine logic.

D.9 SUBROUTINE TDC

Purpose: Multivariable time delay compensation.

Use: CALL TDC(OUT,IN,KPM,TAUM,NTLM,M,N)

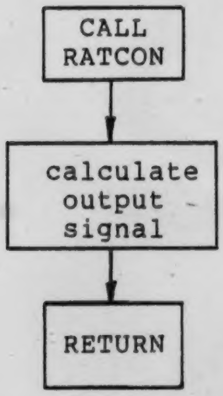


Figure D.14: Subroutine RATCON Program Flow Chart

Parameters: See source code listing in Figure D.15.

Additional Input Data Required from Labelled COMMON:

None.

Brief Description:

The subroutine, TDC, provides a multivariable time delay compensation controller (Ogunnaike and Ray, 1979) for use in DYFLO2 control strategy simulations. Up to 25 time delay compensators can be used in a given simulation program. The algorithm is based on equation (2.92) and the controller structure is given in Figure 2.21. The process is modelled using first order plus time delay transfer functions and the number of input and output variable pairs (i.e. order of the multivariable model) handled by the subroutine is limited only by available computer memory. The process model must be square (i.e. 1x1, 2x2, 3x3, etc.) and a model must be supplied for each block. For blocks where a transfer function does not exist, a model with 0 gain, 0 time delay and 1 for the time constant must be specified to avoid computational problems. The program flow chart given in Figure D.16 will clarify the logic and implementation of the time delay compensation controller.

D.10 SUBROUTINE ZOH

Purpose: Zero order hold.

Use: CALL ZOH(YOUT,XIN,SAMT,TIM,NO)

Parameters: See source code listing in Figure D.17.

Additional Input Data Required from Labelled COMMON:

```
C ********** TDC **********
C
C   TIME DELAY COMPENSATOR SUBROUTINE
C
C   REF.:OGUNNAIKE AND RAY, AIChE J. VOL 25, NO. 6
C        NOVEMBER 1979, P. 1043
C
C   PARAMETERS
C
C   OUT    - OUTPUT SIGNAL VECTOR
C   IN     - INPUT SIGNAL VECTOR
C   KPM    - PROCESS MODEL STEADY STATE GAIN MATRIX
C   TAUM   - PROCESS MODEL TIME CONSTANT MATRIX
C   NTLM   - PROCESS MODEL NUMBER OF TIME LOCATIONS MATRIX
C   M      - DIMENSION OF ABOVE VARIABLES EXACTLY AS IN
C            CALLING ROUTINE
C   N      - ORDER OF PROCESS MODEL, MAX. 25
C
C   INTERNAL VARIABLES
C
C   UDM    - UNDELAYED MODEL OUTPUT
C   YA     - TIME DELAY ROUTINE WORK AREA
C   NTLC   - TIME DELAY LOCATION COUNTER
C   I      - DO LOOP COUNTER
C   JC     - TIME DELAY ROUTINE CALL NUMBER
C   J      - DO LOOP COUNTER
C   DIN    - DUMMY INPUT SIGNAL
C   DTAU   - DUMMY PROCESS MODEL TIME CONSTANT
C   DKP    - DUMMY PROCESS MODEL GAIN
C   UDMD   - DUMMY UNDELAYED MODEL OUTPUT
C   NTLD   - DUMMY NUMBER OF TIME LOCATIONS
C   DMD    - DUMMY DELAYED MODEL OUTPUT
C-------------------------------------------------------------
C
      SUBROUTINE TDC(OUT,IN,KPM,TAUM,NTLM,M,N)
      REAL KPM(M,M),TAUM(M,M),OUT(M),IN(M)
      REAL UDM(25,25)/625*0./,YA(25,500)
      INTEGER NTLM(M,M)
      INTEGER NTLC(25)/25*0/
      JC=0
      DO 30 I=1,N
      OUT(I)=0.
      DO 30 J=1,N
      JC=JC+1
      DIN=IN(J)
      DTAU=TAUM(I,J)
      DKP=KPM(I,J)
      IF(NTLC(JC) .EQ. 0) UDM(I,J)=0.
      CALL TFN1(UDM(I,J),DIN,DTAU,DKP)
      IF(NTLM(I,J) .LE. 0) GOTO 10
      UDMD=UDM(I,J)
      NTLD=NTLM(I,J)+1
      CALL TDL(YA,NTLC,UDMD,DMD,NTLD,JC)
      GOTO 20
   10 DMD=UDMD
   20 OUT(I)=OUT(I)+UDMD-DMD
   30 CONTINUE
      RETURN
      END
```

Figure D.15: Subroutine TDC FORTRAN IV Source Code Listing

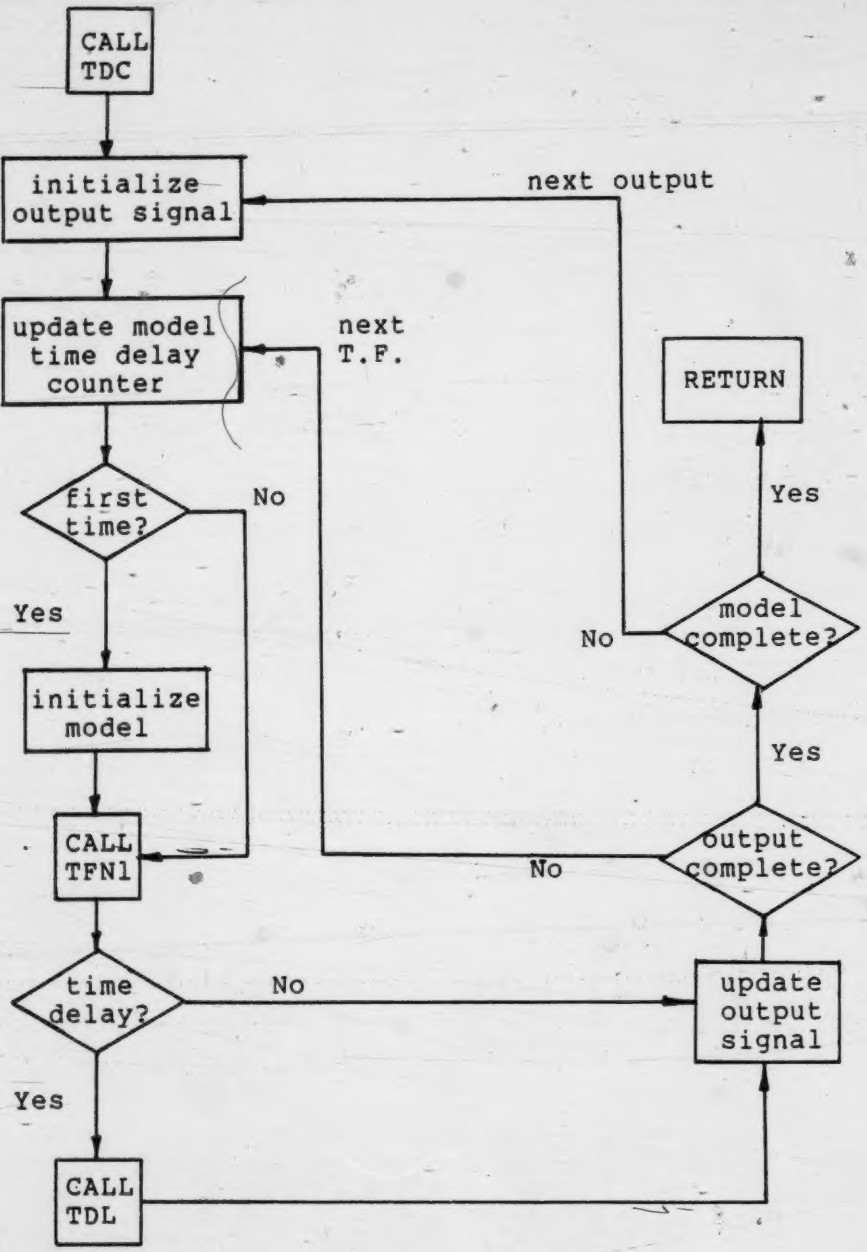

CALL TDC
initialize output signal
next output
update model time delay counter
next T.F.
RETURN
first time?
No
Yes
Yes
initialize model
model complete?
No
Yes
CALL TFN1
output complete?
No
time delay?
No
update output signal
Yes
CALL TDL

Figure D.16: Subroutine TDC Program Flow Chart

```
C ********** ZOH *********************************************
C
C   ZERO ORDER HOLD ROUTINE
C
      SUBROUTINE ZOH(YOUT,XIN,SAMT,TIM,NO)
C
C-------------------------------------------------------------
C
C   PARAMETERS
C
C   YOUT  - OUTPUT SIGNAL
C   XIN   - INPUT SIGNAL
C   SAMT  - SAMPLE TIME
C   TIM   - SIMULATION TIME
C   NO    - ZOH NUMBER - MAX. 10
C
C   INTERNAL
C
C   YHOLD - HOLDS CURRENT OUTPUT SIGNAL
C   STIM  - SIMULATION TIME FOR NEXT SAMPLE
C   TEST  - TAKE NEW SAMPLE FLAG
C   N     - SAMPLE INTERVAL COUNTER
C
C-------------------------------------------------------------
C
      REAL YHOLD(10)
      INTEGER N(10)/10*0/
      IF(TIM .LE. 0.) GOTO 20
      STIM=N(NO)*SAMT
      TEST=TIM-STIM
      IF(TEST) 10,20,20
   10 YOUT=YHOLD(NO)
      GOTO 30
   20 YOUT=XIN
      YHOLD(NO)=YOUT
      N(NO)=N(NO)+1
   30 RETURN
      END
      SUBROUTINE NORM
C
C   THIS SUBROUTINE IS USED WITH INTIMP
C
      RETURN
      END
```

Figure D.17: Subroutine ZOH FORTRAN IV Source Code Listing

None.

Brief Description:

The subroutine, ZOH, provides a technique for simulating the function of a zero order hold for use with sampled data systems. Up to 10 zero order holds may be used in a given simulation program. The program flow chart in Figure D.18 will clarify the logic used.

D.11 SUBROUTINE INTCPD

Purpose: Coupled ordinary differential equation integration subroutine.

Use: CALL INTCPD(N,CMAX,CMIN,X,W)

Parameters: See source code listing in Figure D.19.

Additional Input Data Required from Labelled COMMON:

COMMON/CINT/
IO - integration order

Brief Description:

The subroutine, INTCPD, provides a convenient method for solving a system of up to 30 highly coupled ordinary differential equations. Numerical Jacobian calculations are made and a relaxation factor can be used as well. Figure D.19 shows a source code listing of the subroutine and Figure D.20 presents a program flow chart which will clarify the logic used.

D.12 SUBROUTINE NUMJAC

Purpose: Numerical Jacobian matrix calculation subroutine.

Use: CALL NUMJAC(N,X,XOLD,F,JAC)

Parameters: See source code listing in Figure D.21.

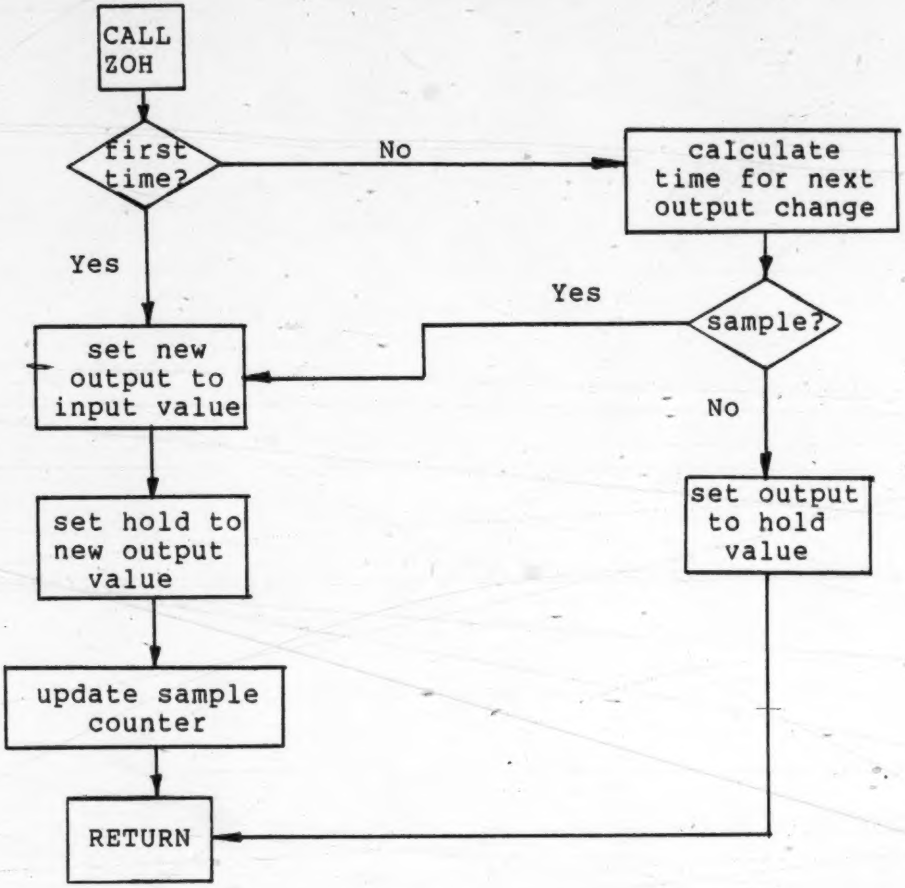

CALL ZOH
first time?
No
calculate time for next output change
Yes
Yes
sample?
set new output to input value
No
set hold to new output value
set output to hold value
update sample counter
RETURN

Figure D.18: Subroutine ZOH Program Flow Chart

```
C*********** INTCPD *********************************************
C
C  COUPLED SYSTEM ODE INTEGRATION ROUTINE
C
C    FOR UP TO 30 ODE'S
C
      SUBROUTINE INTCPD(N,CMAX,CMIN,X,W)
C
C----------------------------------------------------------------
C
C   PARAMETERS
C
C   N  - NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM
C   CMAX - MAXIMUM CHANGE IN INDEPENDENT VARIABLE AS A
C          FRACTION OF THE INDEPENDENT VARIABLE ESTIMATE
C   CMIN - MINIMUM CHANGE IN INDEPENDENT VARIABLE AS A
C          FRACTION OF THE INDEPENDENT VARIABLE ESTIMATE
C   X  - VECTOR CONTAINING INDEPENDENT VARIABLES
C
C-----------------------------------------------------------------
C
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      REAL TKVECT(30),FFVECT(30),X(30),A(30,30),JAC(30,30),DX(30)
      REAL F(30),W(30),CMAX(30),CMIN(30),XOLD(30)
      LOGICAL IFLAG
      TOL1=0.00001
      TOL2=0.00001
C   CALCULATE INVERSE TIME CONSTANT VECTOR
      CALL TKCALC(X,TKVECT)
C   CALCULATE FORCING FUNCTION VECTOR
      CALL FFCALC(X,FFVECT)
C   "PREDICTOR STEP"
      DO 10 I=1,N
C   SAVE PREVIOUS TIME STEP X VECTOR
      XOLD(I)=X(I)
C   INTEGRATE EACH EQUATION IN SET
   10 CALL INTSTF(X(I),TKVECT(I),FFVECT(I))
      ICOUNT=0.
   15 CALL JFUNC(X,XOLD,N,F)
      IFLAG=.TRUE.
      DO 20 I=1,N
C   IF EVERY FUNCTION IS SUFFICIENTLY CLOSE TO ZERO QUIT
C   OTHERWISE PERFORM N-R ITERATION
   20 IF(F(I) .GT. TOL1) IFLAG=.FALSE.
      IF(IFLAG) GOTO 50
C   COMPUTE JACOBIAN MATRIX FOR N-R ITERATION
      CALL NUMJAC(N,X,XOLD,F,JAC)
C   SET UP AUGMENTED "A" MATRIX FOR SUBROUTINE GJR
      DO 30 I=1,N
      A(I,N+1)=-F(I)
      DO 30 J=1,N
   30 A(I,J)=JAC(I,J)
      KEY=0
C   NEWTON - RAPHSON ITERATION
      CALL GJR(A,N,TOL2,KEY,DX,DETER)
C   CHECK FOR SINGULAR SYSTEM AND PRINT WARNING
      IF(ABS(DETER) .LT. TOL2) WRITE(6,1) DETER
    1 FORMAT(//,'WARNING!!!  JACOBIAN SINGULAR IN ROUTINE INTCPD',
     +/,'   DETERMINANT IS  ',F12.6,/)
C   RELAX SOLUTION VECTOR
      CALL RELAX(N,X,DX,CMAX,CMIN,W)
      DO 40 I=1,N
   40 X(I)=X(I)+W(I)*DX(I)
      ICOUNT=ICOUNT+1
C   MAXIMUM NUMBER OF N-R ITERATIONS ALLOWED
      IF(ICOUNT .GE. 100) GOTO 45
      GOTO 15
   45 WRITE(6,2) ICOUNT
    2 FORMAT(//,'WARNING!!! N-R ITERATION IN ROUTINE INTCPD DID ',
     +'NOT CONVERGE IN ',I5,' ITERATIONS.',/)
   50 RETURN
      END
```

Figure D.19: Subroutine INTCPD FORTRAN IV Source Code Listing

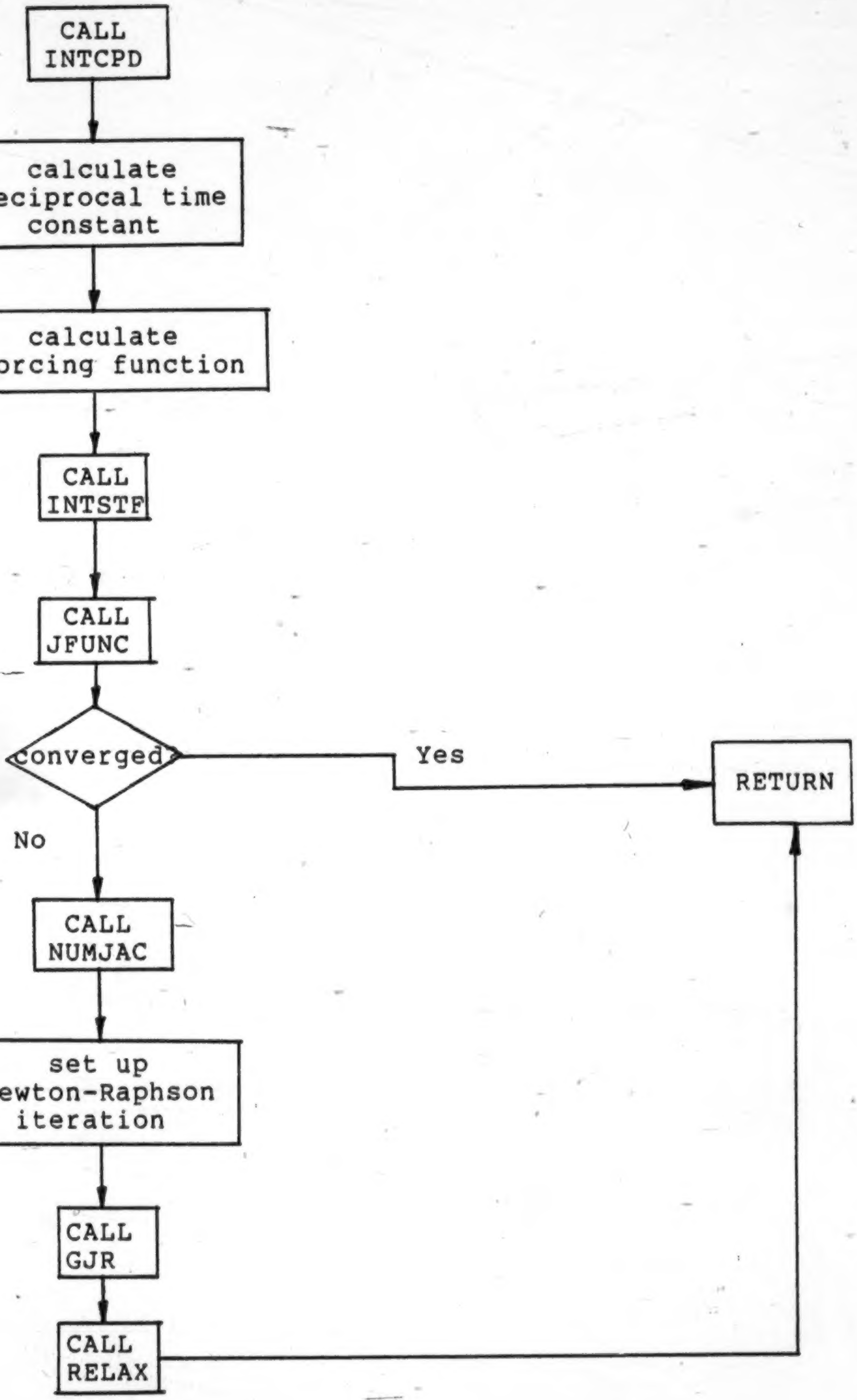

CALL INTCPD
calculate reciprocal time constant
calculate forcing function
CALL INTSTF
CALL JFUNC
converged?
Yes
RETURN
No
CALL NUMJAC
set up Newton-Raphson iteration
CALL GJR
CALL RELAX

Figure D.20: Subroutine INTCPD Program Flow Chart

```
C********** NUMJAC *******************************************
C
C  NUMERICAL JACOBIAN CALCULATION ROUTINE
C
      SUBROUTINE NUMJAC(N,X,XOLD,F,JAC)
C
C------------------------------------------------------------
C
C   PARAMETERS
C
C   N  - NUMBER OF EQUATIONS IN SYSTEM
C   X  - INDEPENDENT VARIABLE VECTOR
C   XOLD - PREVIOUS TIME STEP INDEPENDENT VARIABLE VECTOR
C   F  - FUNCTION VALUE VECTOR
C   JAC - JACOBIAN MATRIX
C
C-------------------------------------------------------------
C
      REAL X(30),F(30),JAC(30,30),XNEW(30),XP(30),FNEW(30)
      REAL XOLD(30)
      PER=0.01
      DO 10 J=1,N
C   PERTURB SOLUTION VECTOR AND
C   ASSURE NON-ZERO ELEMENTS
      XP(J)=PER*X(J)
   10 IF(XP(J) .LT. 0.000001) XP(J)=0.000001
C   COMPUTE EACH COLUMN OF JACOBIAN MATRIX
      DO 30 J=1,N
      DO 20 I=1,N
      XNEW(I)=X(I)
   20 IF(I .EQ. J) XNEW(I)=X(I)+XP(I)
      CALL JFUNC(XNEW,XOLD,N,FNEW)
      DO 30 I=1,N
   30 JAC(I,J)=(FNEW(I)-F(I))/XP(J)
      RETURN
      END
```

Figure D.21: Subroutine NUMJAC FORTRAN IV Source Code Listing

Additional Input Data Required from Labelled COMMON:
None.

Brief Description:

The subroutine, NUMJAC, provides an easy method of computingf the Jacobian matrix for a system of coupled ordinary differential equations. Figure D.21 gives the source code listing of the subroutine and Figure D.22 is a program flow chart which will clarify the logic used.

D.13 SUBROUTINE JFUNC

Purpose: Function evaluation subroutine for use in conjunction with NUMJAC

Use: CALL JFUNC(X,XOLD,N,F)

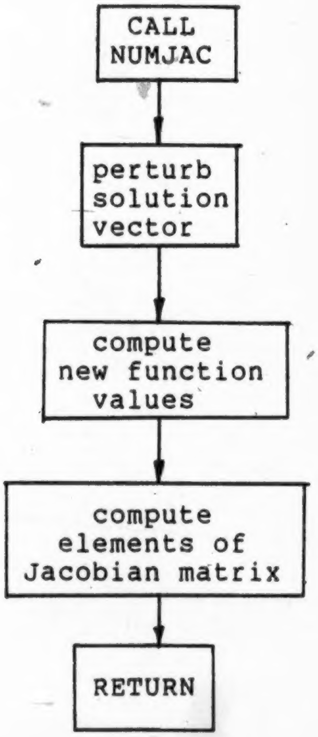

CALL NUMJAC
perturb solution vector
compute new function values
compute elements of Jacobian matrix
RETURN

Figure D.22: Subroutine NUMJAC Program Flow Chart

Parameters: See source code listing in Figure D.23.

Additional Input Data Reqwuired from Labelled COMMON:
None.

Brief Description:

The subroutine, JFUNC, provides a convenient method of evaluating the Jacobian functions for a set of ODE's. Figure D.23 presents a source code listing of tghe subroutine and Figure D.24 gives a program flow chart.

D.14 SUBROUTINE RELAX

Purpose: Smart relaxation factor subroutine.

Use: CALL RELAX(N,X,DX,CMAX,CMIN,W)

Parameters: See source code listing in Figure D.25.

Additional Input Data Required from Labelled COMMON:
None.

Brief Description:

The subroutine, RELAX, provides a simple method to control and adapt the relaxation factor vector based on the rate of change in the independent variable. It is capable of either speeding up or slowing down the solution convergence. Figure D.25 shows a source code listing of the subroutine and Figure D.26 presents a program flow chart which will clarify the logic used.

```
C*********** JFUNC ***********************************************
C
C  FUNCTION EVALUATION ROUTINE
C
      SUBROUTINE JFUNC(X,XOLD,N,F)
C
C-----------------------------------------------------------------
C
C   PARAMETERS
C   X  - INDEPENDENT VARIABLE VECTOR
C   XOLD - PREVIOUS INDEPENDENT VARIABLE VECTOR
C   N  - NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM
C   F  - FUNCTION VALUE VECTOR
C
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      REAL X(30),F(30),XOLD(30),TKVECT(30),FFVECT(30)
C   COMPUTE MOST RECENT INVERSE TIME CONSTANT VECTOR
      CALL TKCALC(X,TKVECT)
C   COMPUTE MOST RECENT FORCING FUNCTION VECTOR
      CALL FFCALC(X,FFVECT)
C   EVALUATE FUNCTION VALUES
      DO 10 I=1,N
   10 F(I)=X(I)*(1.+TKVECT(I)*DT)-(FFVECT(I)*DT+XOLD(I))
      RETURN
      END
```

Figure D.23: Subroutine JFUNC FORTRAN IV Source Code Listing

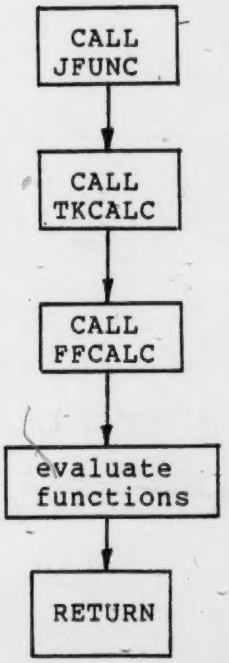


Figure D.24: Subroutine JFUNC Program Flow Chart

```
C********** RELAX *******************************************
C
C
C  SMART RELAXATION FACTOR CALCULATION ROUTINE
C
C
      SUBROUTINE RELAX(N,X,DX,CMAX,CMIN,W)
C
C-------------------------------------------------------------
C
C   PARAMETERS
C
C   N  - NUMBER OF ORDINARY DIFFERENTIAL EQUATIONS IN SYSTEM
C   X  - INDEBENDENT VARIABLE VECTOR
C   DX - CHANGE IN INDEPENDENT VARIABLE VECTOR FROM N-R ITERATION
C   CMAX - MAXIMUM CHANGE IN INDEPENDENT VARIABLE
C   CMIN - MINIMUM CHANGE IN INDEPENDENT VARIABLE
C   W  - RELAXATION FACTOR
C-------------------------------------------------------------
C
      REAL X(30),DX(30),CMAX(30),CMIN(30),W(30)
      DO 60 I=1,N
C   AVOID DIVIDE BY ZERO IN CHANGE CALCULATION
      IF(ABS(X(I)) .LT. 0.00001) GOTO 50
C   CALCULATE INDEPENDENT VARIABLE CHANGE
      CHANGE=ABS(DX(I)/X(I))
C  UNDER RELAX
      IF(CHANGE .GT. CMAX(I)) GOTO 10
C  OVER RELAX
      IF(CHANGE .LT. CMIN(I)) GOTO 30
C  NO RELAXATION
      GOTO 50
C  CALCULATE UNDER RELAXATION FACTOR
   10 AA=(ABS(DX(I))-CMAX(I))/CMAX(I)
      IF(AA .GT. 0.99) GOTO 20
      W(I)=1.0-AA
      GOTO 60
C  MINIMUM RELAXATION FACTOR
   20 W(I)=0.01
      GOTO 60
C  CALCULATE OVER RELAXATION FACTOR
   30 BB=(CMIN(I)-ABS(DX(I)))/CMIN(I)
      IF(BB .GT. 0.99) GOTO 40
      W(I)=1.0+BB
      GOTO 60
C MAXIMUM RELAXATION FACTOR
   40 W(I)=1.99
      GOTO 60
   50 W(I)=1.0
   60 CONTINUE
      RETURN
      END
```

Figure D.25: Subroutine RELAX FORTRAN IV Source Code Listing

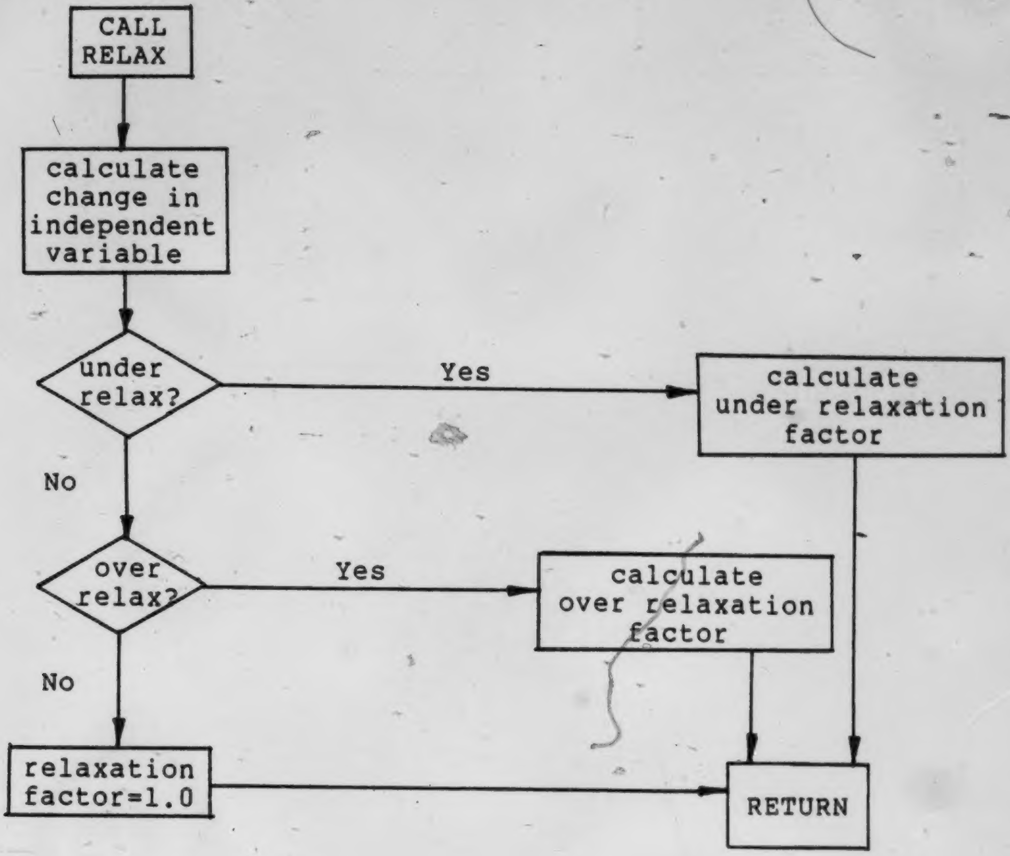

CALL RELAX
calculate change in independent variable
under relax?
Yes
calculate under relaxation factor
No
over relax?
Yes
calculate over relaxation factor
No
relaxation factor=1.0
RETURN

Figure D.26: Subroutine RELAX Program Flow Chart

Appendix E: Lead/Lag Decoupling Analysis

E.1 Lead/Lag Compensator - Subroutine LLCOMP

The FORTRAN implementation of the compensator described in section 2.5 is given in Figure D.8. The code in this figure is easily understood when combined with the discussion in section 2.6.4. The subroutine performs all the calculations for a first order lead/lag unit plus a time delay and may be used up to 25 times in a given simulation. This routine is written to be compatible with DYFLO2 and is placed in the control subroutine group.

To verify that the routine functions correctly, some open loop simulations were performed and the computed results compared with values calculated from the analytical solution. The response of both a lead time dominant compensator and a lag time dominant compensator were examined. The forcing function used in both cases was a step increase in the input variable.

The analytical solution for a step input to a lead/lag compensator is easily established. For the lead/lag compensator expressed by the transfer function:

$$G(s)=\frac{Y(s)}{X(s)}=\frac{(\tau_1 s+1)K_{LL}}{\tau_2 s+1} \qquad (E.1)$$

for a step input:

$$X(s)=\frac{K_s}{s} \qquad (E.2)$$

it follows that substituting for X(s) in equation (E.1) using equation (E.2) gives:

$$Y(s)=\frac{K_{LL}K_s(\tau_1 s+1)}{s(\tau_2 s+1)} \qquad (E.3)$$

Inversion of equation (E.3) from the Laplace domain to the time domain yields:

$$y(t)=K_{LL}K_s\left[\frac{\tau_1-\tau_2}{\tau_2}e^{-t/\tau_2}+1\right] \qquad (E.4)$$

Equation (E.4) is the general analytical solution for a lead/lag compensator subjected to a step forcing function used in verification of the simulated results.

The test compensators used were:

Lead Time Dominant

$$G_{LD}(s)=\frac{0.5(40.0s+1)}{(5.0s+1)} \qquad (E.5)$$

Lag Time Dominant

$$G_{LG}(s)=\frac{0.5(5.0s+1)}{(40.0s+1)} \qquad (E.6)$$

These two equations correspond to loop 1 of the idealized decoupling exercise described in section E.2. For simplicity, the step change magnitude was chosen to be unity (i.e. $K_s=1$).

The analytical and numerical solutions for both compensators are given in Figures E.1 and E.2,

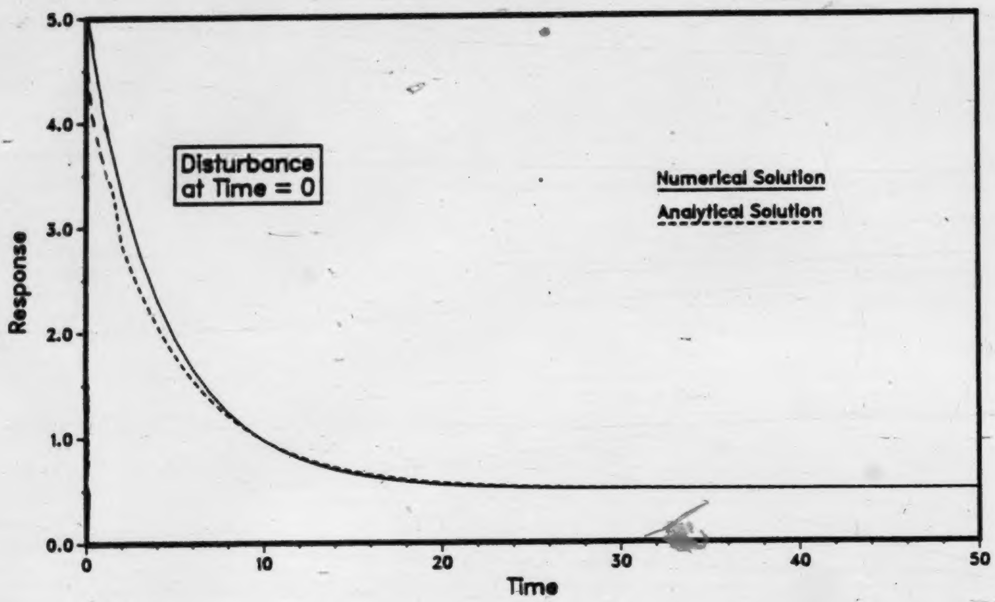


Figure E.1: Lead Time Dominant Lead-Lag Compensator Response to a Step Input

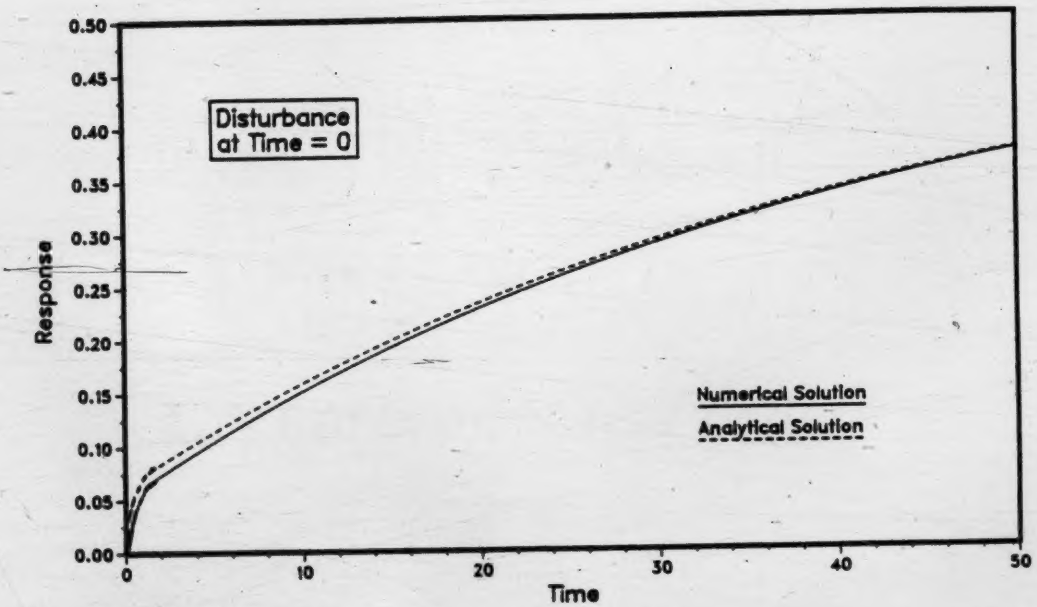


Figure E.2: Lag Time Dominant Lead-Lag Compensator Response to a Step Input

respectively. The numerical results were obtained using an integration interval of 1.0 time unit. Although there is an excellent correspondence between these solutions for both compensators, even closer agreement of the numerical results with the analytical solution can be obtained using smaller integration intervals.

A time delay can be added to the compensator by simply cascading it with a delay term. As shown in line 61 in Figure D.8, this is implemented via the DYFLO2 routine "TDL". Figure E.3 presents the response of the compensator in equation (E.6) for a 2 time unit delay for the same unit step change in input. It can be seen from this figure that

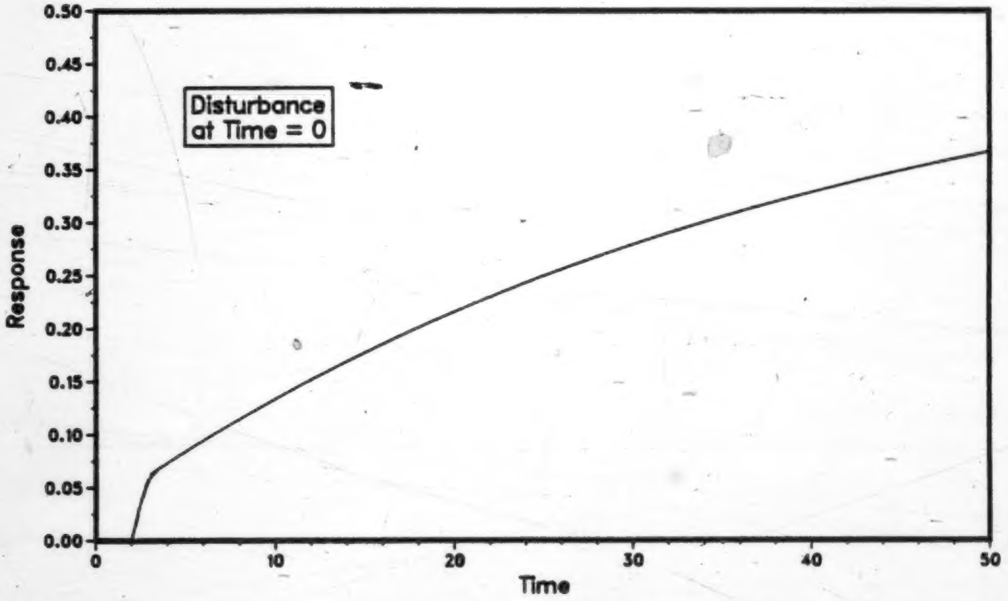


Figure E.3: Lag Time Dominant Lead-Lag Compensator with a Two Unit Delay

the numerical response given in Figure E.2 has been delayed in time by two units. The source code of the driver program used for these simulations is given in Figure E.4.

E.2 Decoupling (Noninteracting) Control

E.2.1 Introduction

To examine the behaviour of decoupling controllers in various situations, several simulations under somewhat idealized conditions were performed. The 2x2 plant transfer function matrix used for these simulations was:

$$G_p(s) = \begin{bmatrix} \dfrac{2e^{-1s}}{5s+1} & \dfrac{1e^{-3s}}{40s+1} \\ \dfrac{1e^{-2s}}{30s+1} & \dfrac{2e^{-1s}}{20s+1} \end{bmatrix} \qquad (E.7)$$

and the load transfer function vector was:

$$G_l(s) = \begin{bmatrix} \dfrac{1}{10s+1} \\ \dfrac{1}{15s+1} \end{bmatrix} \qquad (E.8)$$

Each of the elements in equations (E.7) and (E.8) correspond to the appropriate block in Figure 2.12. It should be noted that the direct transmission transfer function associated with y_1 has a relatively small time constant when compared to the interacting (cross) transfer functions for either output variable. Although the direct transmission transfer function associated with y_2 has a larger time constant than the first, it is still smaller than either interaction time

```
C
C     LEAD LAG DECOUPLING TEST DRIVER
C
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4
      COMMON/SIM/ SIMT,RINT,PT,TIM
      REAL KP(2,2),KL(2),TAU(2,2),TAUL(2),DX(2),DY(2),DD(2),YA(25,500)
      REAL KC1,KC2,KLL1,KLL2,D(2,2)
      INTEGER NTL(2,2),NTLL(2)
      COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
      LOGICAL NF
C
C     INITIALIZE SYSTEM VARIABLES
C
      NF=.FALSE.
      TIM=0.
      RINT=0.1
      SIMT=100.
      PT=1.
      DISTIM=1.99
      IO=1
      JN=0
      JS=0
      JS4=0
C
C     INITIALIZE LINEAR MODEL PARAMETERS
C
      CALL LMOD(RINT)
      CO11=0.
      CO22=0.
      DCO11=0.
      DCO22=0.
      CO12=0.
      CO21=0.
      KLL1=-KP(1,2)/KP(1,1)
      KLL2=-KP(2,1)/KP(2,2)
      SP1=50.
      SP2=50.
      KC1=2.29
      KC2=9.04
      TI1=2.35
      TI2=3.015
      PB1=100./KC1
      PB2=100./KC2
      RPT1=1./TI1
      RPT2=1./TI2
      ZR1=0.
      ZR2=0.
      FSRDG1=100.
      FSRDG2=100.
      AXN1=-1.
      AXN2=-1.
      OI1=50.
      OI2=50.
      SSY1=50.
      SSY2=50.
      Y1=SSY1
      Y2=SSY2
      DLOAD=0.
      D(1,1)=0.666
      D(1,2)=-0.333
      D(2,1)=-0.333
      D(2,2)=0.666
      TD1=RINT*FLOAT(NTL(1,2)-NTL(1,1))
      TD2=RINT*FLOAT(NTL(2,1)-NTL(2,2))
      IF(TD1 .LE. 0.) TD1=0.
      IF(TD2 .LE. 0.) TD2=0.
      WRITE(6,1)
    1 FORMAT(/,5X,'LEAD - LAG COMPENSATOR TRANSFER FUNCTIONS',//)
      WRITE(6,2) TAU(1,1),TAU(2,2)
    2 FORMAT(19X,'(',F7.3,'s+1)',32X,'(',F7.3,'s+1)')
      WRITE(6,3) KLL1,TD1,KLL2,TD2
    3 FORMAT(7X,'G12(s)=',F5.2,12('-'),'*exp(-',F5.2,'s)',
     +       7X,'G21(s)=',F5.2,12('-'),'*exp(-',F5.2,'s)')
      WRITE(6,4) TAU(1,2),TAU(2,1)
    4 FORMAT(20X,F7.3,'s+1',34X,F7.3,'s+1',//)
      WRITE(6,5)
    5 FORMAT(4X,'TIME',7X,'DCO11',8X,'CO12',6X,'INPUT 1 ',4X,
     +'OUTPUT 1',6X,'DCO22',8X,'CO21',6X,'INPUT 2 ',4X,'OUTPUT 2',/)
C
C     MAIN SIMULATION LOOP
C
C     DERIVATIVE SECTION
C
C     DISTURBANCE
C
   10 IF(TIM .GE. DISTIM) SP1=55.
      DD(1)=DLOAD
      DD(2)=DLOAD
      IF(TIM .GE. DISTIM) PT=.1
      IF(TIM .GE. (DISTIM*10.)) PT=1.
C
C     TEST FOR PRINT AND FINISH
C
      CALL PRNTF(PT,SIMT,NF,TIM,DCO11,CO12,DX(1),DY(1),DCO22,
     +     CO21,DX(2),DY(2),DLOAD)
      IF(NF) GOTO 20
C
C     INTEGRATION SECTION
C
      CALL INTI(TIM,RINT,IO)
C
C     CONTROLLER CALCULATIONS
C
      CI1=Y1
      CALL CONTR2(CI1,CO11,ZR1,FSRDG1,SP1,AXN1,PB1,RPT1,OI1)
      IF(ABS(CO11) .LT. 1.0E-10) CO11=0.
      CI2=Y2
      CALL CONTR2(CI2,CO22,ZR2,FSRDG2,SP2,AXN2,PB2,RPT2,OI2)
      IF(ABS(CO22) .LT. 1.0E-10) CO22=0.
      DCO11=CO11-SSY1
      DCO22=CO22-SSY2
C
C     STATIC DECOUPLING
C
C      TURN OFF WITH COMMENT CARDS
C
      CO12=DCO22*D(1,2)
      CO21=DCO11*D(2,1)
      DCO11=DCO11*D(1,1)
      DCO22=DCO22*D(2,2)
C
C     DYNAMIC DECOUPLING
C     TURN OFF WITH COMMENT CARDS
C
C       CALL LLCOMP(CO12,DCO22,TAU(1,1),TAU(1,2),TD1,KLL1,1)
C       CALL LLCOMP(CO21,DCO11,TAU(2,2),TAU(2,1),TD2,KLL2,2)
C
C     SET DECOUPLER OUTPUT TO 0 IF NO DECOUPLING
C       (TURN ON OR OFF WITH COMMENT CARDS)
C
```

Figure E.4: Lead-Lag Decoupling DYFLO2 Program Source Code Listing

```
C       CO12=0.
C       CO21=0.
C
C    PLANT CALCULATIONS
C
C
C    PLANT INPUTS
C
      DX(1)=DCO11+CO12
      DX(2)=DCO22+CO21
      CALL I2BY2(TIM,RINT,1)
      Y1=DY(1)+SSY1
      Y2=DY(2)+SSY2
      GOTO 10
   20 STOP
      END
C
C-------------------------------------------------------------
C
C    2x2 COUPLED SYSTEM PARAMETER INITIALIZATION ROUTINE
C
C-------------------------------------------------------------
C
      SUBROUTINE LMOD(RINT)
      REAL KP(2,2),KL(2),TAU(2,2),TAUL(2),DX(2),DY(2),DD(2),YA(25,500)
      INTEGER NTL(2,2),NTLL(2)
      COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
C
C    INITIALIZE TRANSFER FUNCTION MODEL PARAMETERS
C
C---$ GAINS
C
      KP(1,1)=2.
      KP(2,1)=1.
      KP(1,2)=1.
      KP(2,2)=2.
      KL(1)=1.
      KL(2)=1.
C
C---$ TIME CONSTANTS
C
      TAU(1,1)=5.
      TAU(2,1)=30.
      TAU(1,2)=40.
      TAU(2,2)=20.
      TAUL(1)=10.
      TAUL(2)=15.
C
C---$ DEAD TIME
C
      TD11=1.
      TD21=2.
      TD12=3.
      TD22=1.
      TDLD1=0.
      TDLD2=0.
C
C---$ S.S. OPERATING CONDITIONS
C
      DX(1)=0.
      DX(2)=0.
      DY(1)=0.
      DY(2)=0.
C
C    CALCULATE NUMBER OF INTEGRATION STEPS IN TIME DELAYS
C
      NTL(1,1)=INT(TD11/RINT)
      NTL(2,1)=INT(TD21/RINT)
      NTL(1,2)=INT(TD12/RINT)
      NTL(2,2)=INT(TD22/RINT)
      NTLL(1)=INT(TDLD1/RINT)
      NTLL(2)=INT(TDLD2/RINT)
      RETURN
      END
```

Figure E.4 continued

constant. Also, the time delay elements of each transfer function were chosen to ensure controller realizability when Zalkind's (1967) design procedure is used.

Simulations using both static and dynamic decoupling techniques on the plant model in equations (E.7) and (E.8) were performed in addition to studying the behavior of the original undecoupled system. The only dynamic decoupling method investigated was that of Zalkind because not only is it similar to Boksenbom and Hood's (1949) when implemented, but also, the ideal decoupling controller described by Luyben (1970) only increases the decoupler complexity without providing any additional benefit.

Several other simulations were performed using various different forms of the process model in equations (E.7) and (E.8). Each variation will be discussed in conjunction with the appropriate case.

E.2.2 Interaction Analysis

Before proceeding to the controller or decoupler design, it is useful to analyze the interactions of the plant. For this system, the Relative Gain Matrix (RGM) can be found analytically. Assuming a unit step input change in both inputs (independently of course), the open loop gains can be computed from the limit as s approaches zero of each element of equation (E.7). The delay terms are ignored as well. Equation (2.59) then gives:

$$A=\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix} \qquad (E.9)$$

The transposed inverse of the above equation is:

$$B= \begin{bmatrix} 0.667 & -0.333 \\ -0.333 & 0.667 \end{bmatrix} \qquad (E.10)$$

The RGM, from equation (2.54), is:

$$RGM= \begin{array}{c|cc} & u_1 & u_2 \\ \hline y_1 & 1.333 & -0.333 \\ y_2 & -0.333 & 1.333 \end{array} \qquad (E.11)$$

It is not by chance that the pairing is u_1 with y_1 and u_2 with y_2!

E.2.3 Controller Design

Proportional plus integral (PI) controllers were arbitrarily used in the main loops (G_{C11} and G_{C22}) for all simulations. Tuning the controllers can be accomplished by performing open loop step response simulations as outlined in section 5.5. However, because the exact process transfer functions are known in this case, the controller constants can be determined based on the diagonal elements of equation (E.7) and equations (5.21) and (5.22). Although this method will not give the "best" constants because of the presence of off-diagonal transfer functions in the process transfer function matrix model, the estimates will be adequate for this investigation. The two sets of PI controller constants used in all simulations discussed in this appendix unless otherwise noted were:

$K_{p1} = 2.29$

$\tau_{I1} = 2.35$

and:

$K_{p2} = 9.04$

$\tau_{I2} = 3.02$

E.2.4 Multi-loop PI control of the Original System

Figure E.5 shows the undecoupled system response to a 5% step increase in the setpoint in loop 1. It can be seen from this figure that this disturbance has created an upset in loop 2! This response can be improved by "decoupling" the plant.

E.2.5 Static Decoupling of the Original System

As a starting point, static decoupling of the original process was attempted. This is the simplest approach and easily implemented. Using the steady state process gain matrix (equation E.9), it follows that the static decoupling controller is:

$$D= \begin{bmatrix} 0.667 & -0.333 \\ -0.333 & 0.667 \end{bmatrix} \tag{E.12}$$

Each element in the above equation corresponds to the appropriate block in the block diagram shown in Figure 2.15.

The decoupled system response to a 5% step in the loop 1 setpoint is shown in Figure E.6. As expected, the overall system response has been improved.

E.2.6 Dynamic Decoupling of the Original System

It is intuitively obvious that dynamic decoupling has the potential for offering greater improvements in control

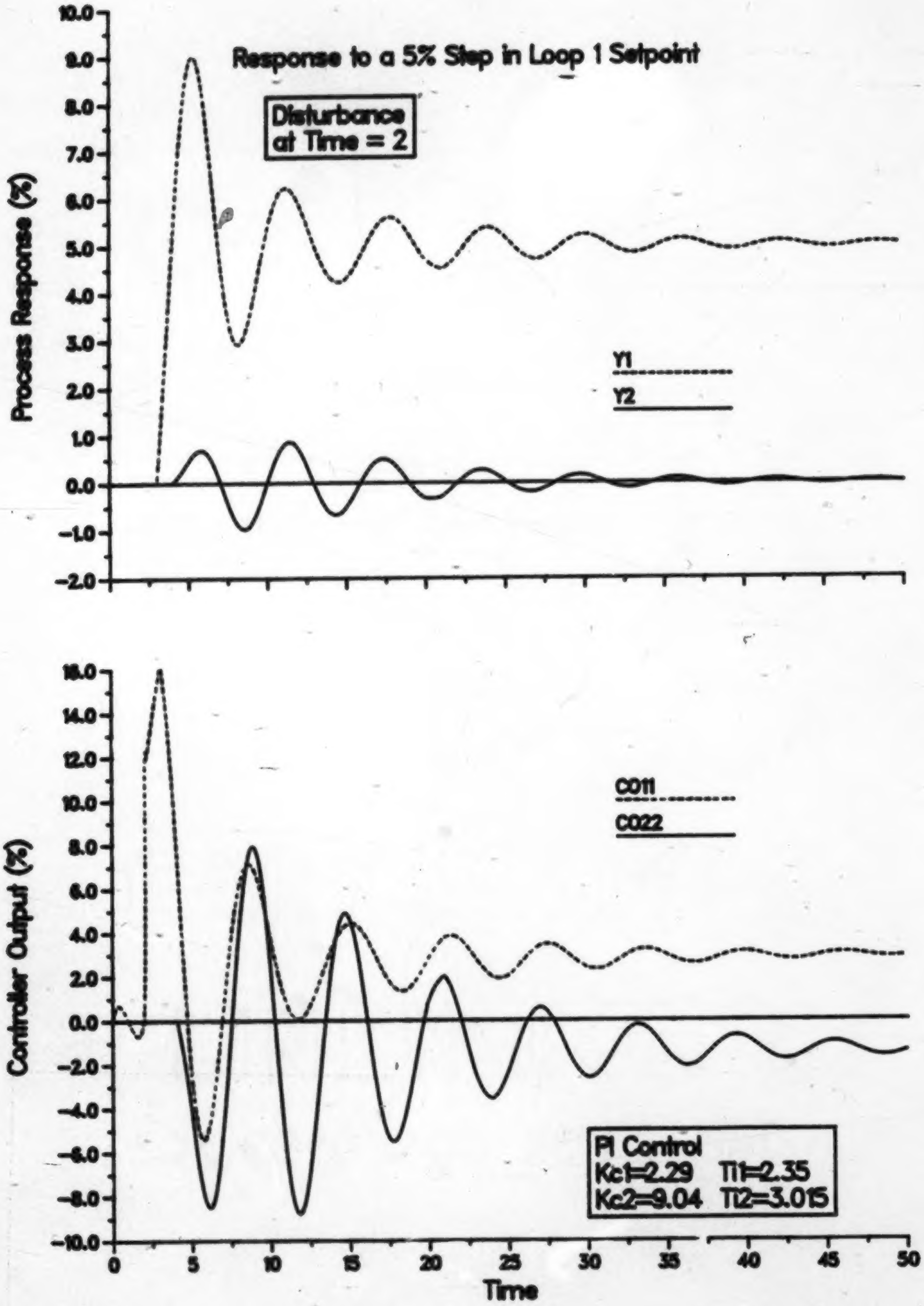


Figure E.5: Multiloop PI Control of the Original Process

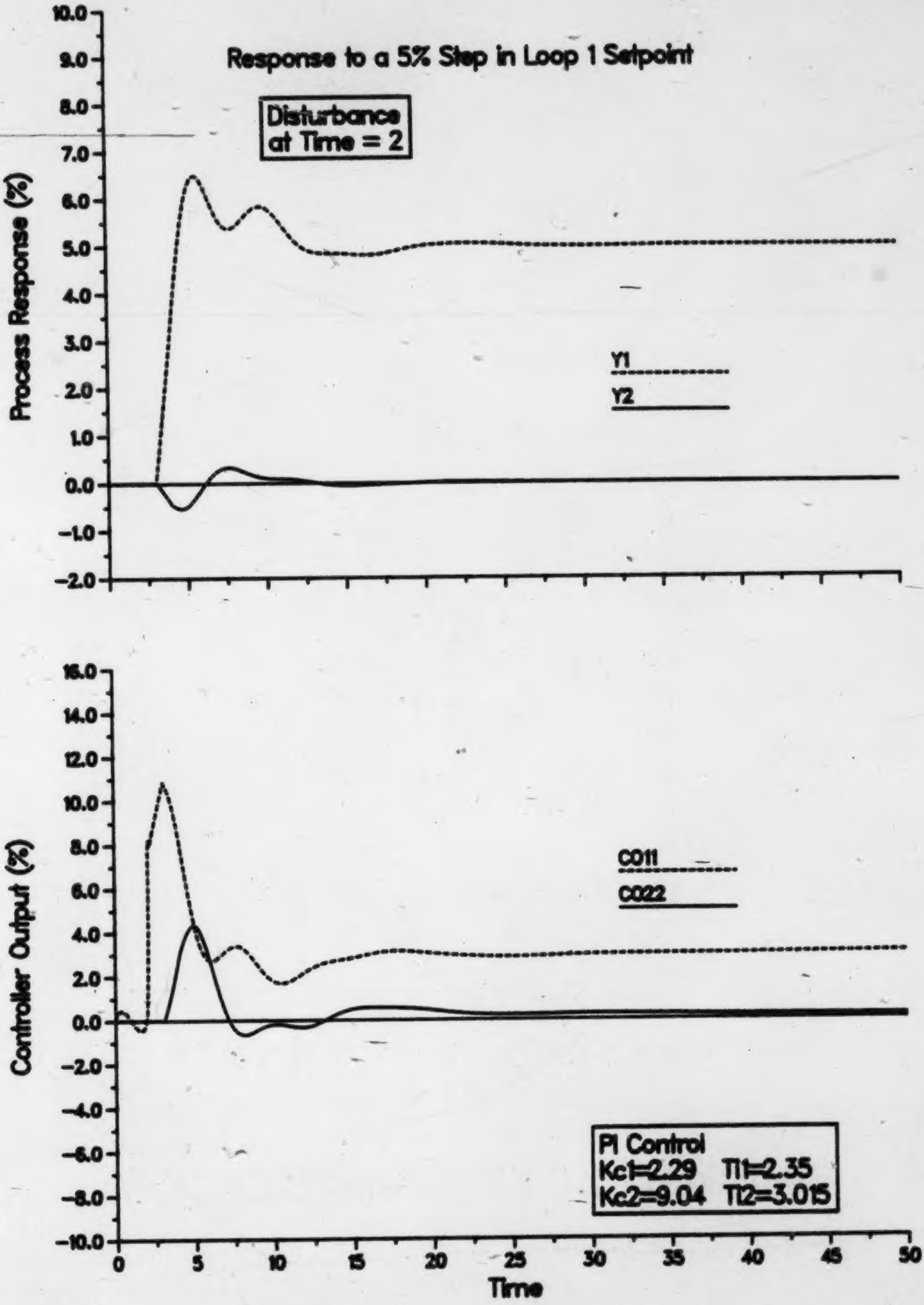


Figure E.6: Multiloop PI Control of the Original Process with Static Decoupling

system performance given the penalty of a more complex decoupler. From equations (2.80) and (2.81), the decoupling controllers for the process model given in equation (E.7), are found to be first order lead/lag compensators followed by a delay term of the form:

$$G^*_{C12}=\frac{-0.5(5.0s+1)e^{-2s}}{(40.0s+1)} \quad (E.13)$$

and:

$$G^*_{C21}=\frac{-0.5(20.0s+1)e^{-1s}}{(30.0s+1)} \quad (E.14)$$

For the sake of demonstration, no delays are included for the first dynamic decoupling simulation. Perfect decoupling as observed by Zalkind is thus expected. Inserting these decouplers into the system as shown in Figure 2.18 improves the performance tremendously! The decoupled system response to the 5% step in the loop 1 setpoint is shown in Figure E.7. The response of the coupled system without time delays is given in Figure E.8 for comparison with Figure E.7. It can be seen that the interaction with loop 2 been completely eliminated.

More realistically, the delay terms in the process model should be included. Figure E.9 presents the simulation response of the original model with the decoupling controllers using the same disturbance. The overall system stability and decoupling performance has deteriorated somewhat from the response in Figure E.7. However, when Figure E.9 is compared with the original undecoupled

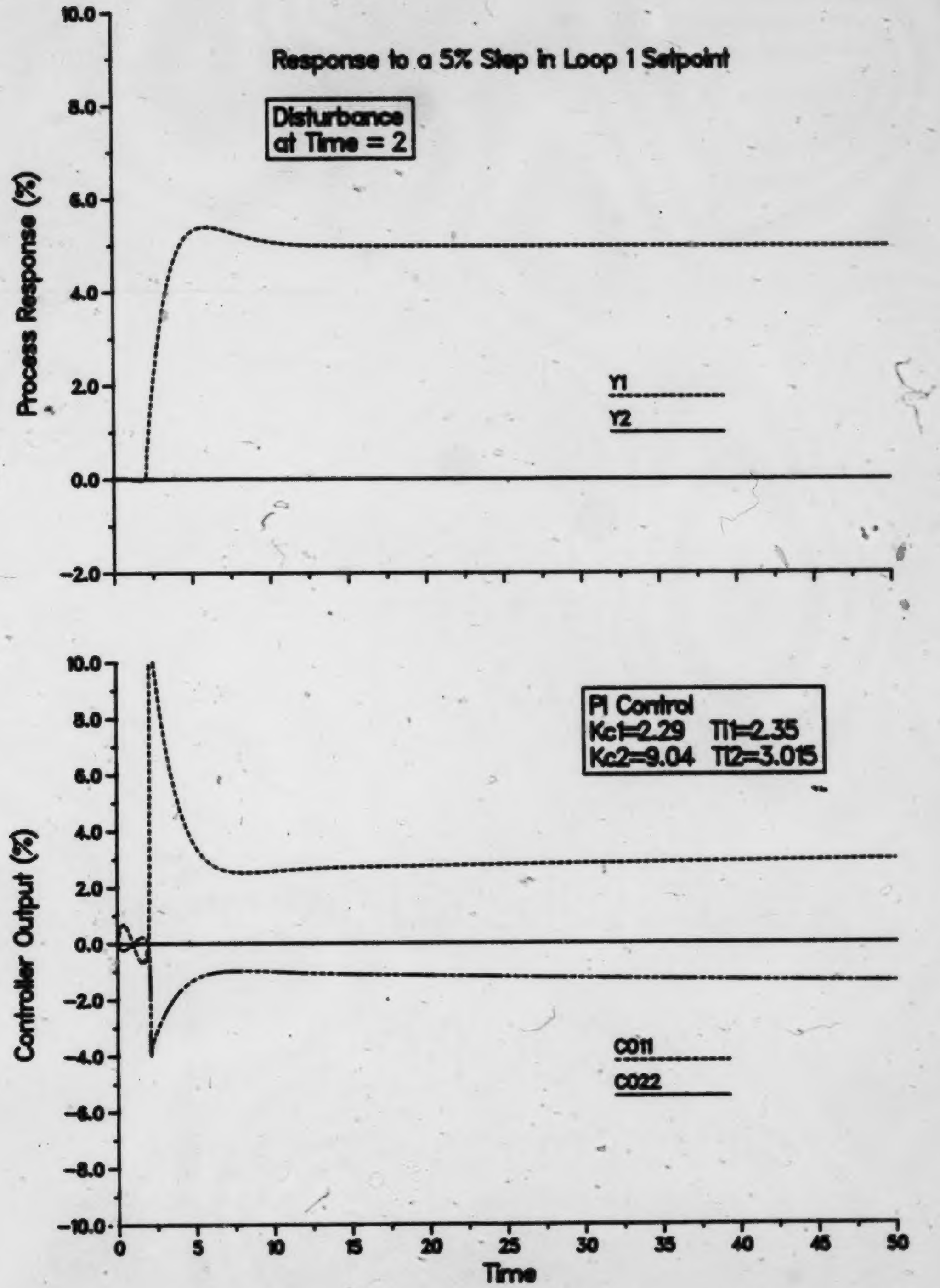


Figure E.7: Multiloop PI Control of the Undelayed Process with Dynamic Decoupling

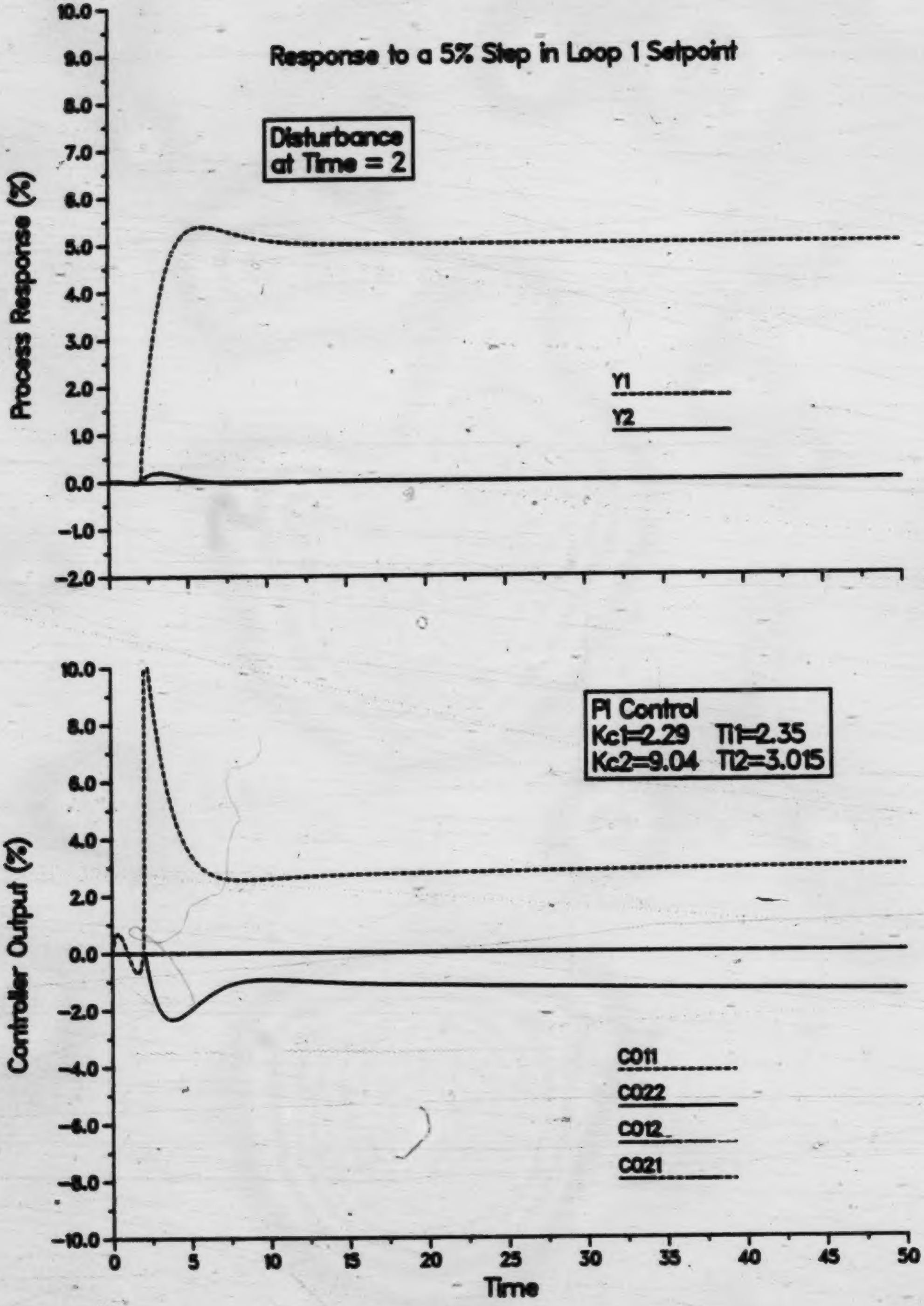


Figure E.8: Multiloop PI Control of the Undelayed Process

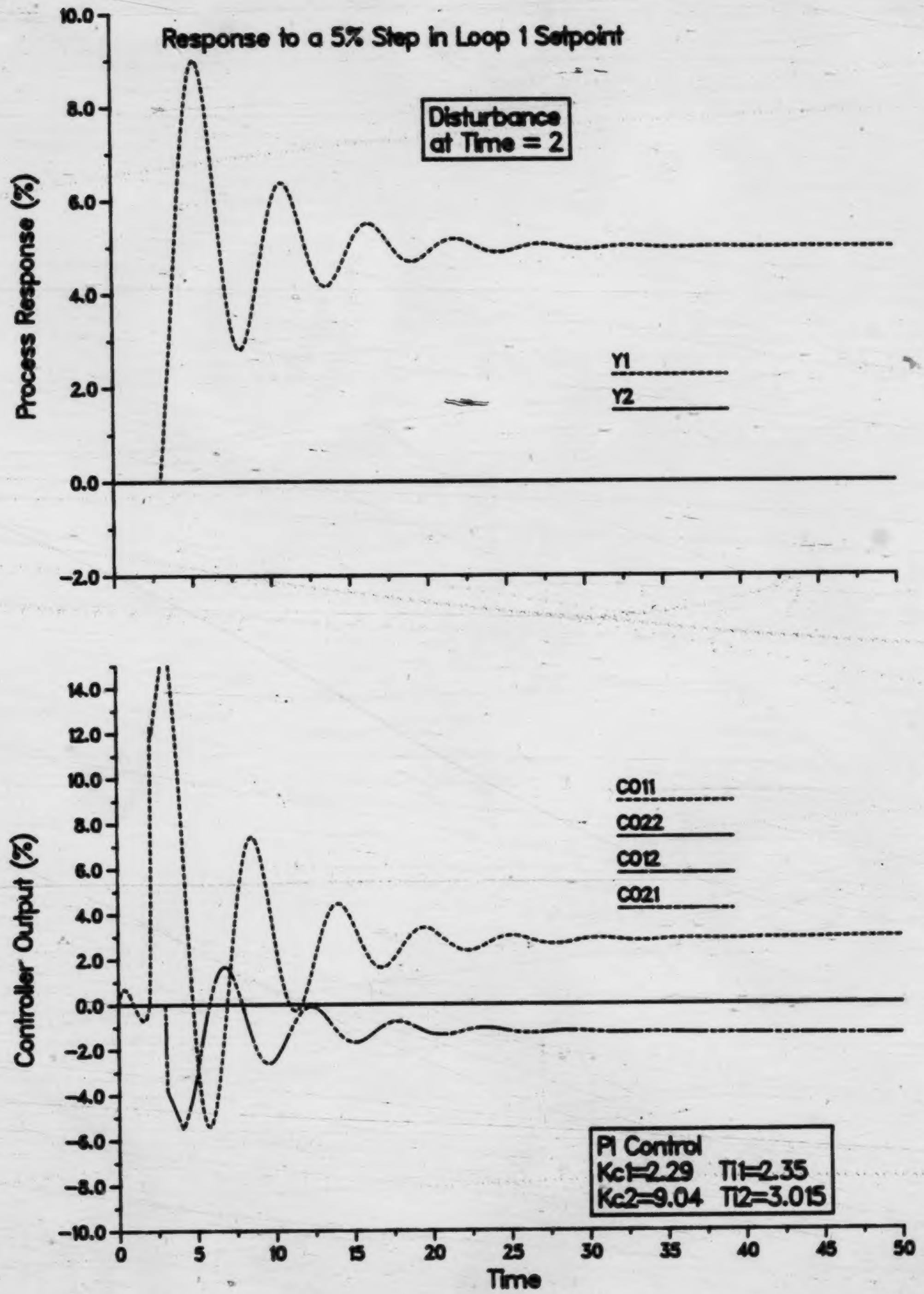


Figure E.9: Multiloop PI Control of the Original Process with Dynamic Decoupling

response given in Figure E.5, the benefits associated with decoupling become clear. It should be noted that this difference in behavior was expected due to the destabilizing effect process time delays have on control systems. From these results, it is felt that additional performance improvements could be realized if time delay compensation were incorporated into the control system. This is the subject of Appendix F.

For completeness, Figure E.10 presents the response of the decoupled system (including time delays) to a 10% load increase. Both loops handle the disturbance with a reasonable amount of success.

E.2.7 Decoupled Multi-loop PI Control with Unrealizable Decoupling Predictions

Next, an unrealizable decoupling controller situation was considered. The process model used was the same as equation E.7 except the delay terms in the second row were interchanged. Obviously, this results in a decoupler for loop 2 that is predictive in nature. That is, the decoupler given in equation (E.14) becomes:

$$G^*_{C21} = \frac{-0.5(20.0s+1)e^{+1s}}{(30.0s+1)} \qquad \text{(E.15)}$$

The exponential term in this equation requires prediction one time unit into the future which is not possible so the prediction term on the decoupler is simply dropped. Figure E.11 shows the response for this form of

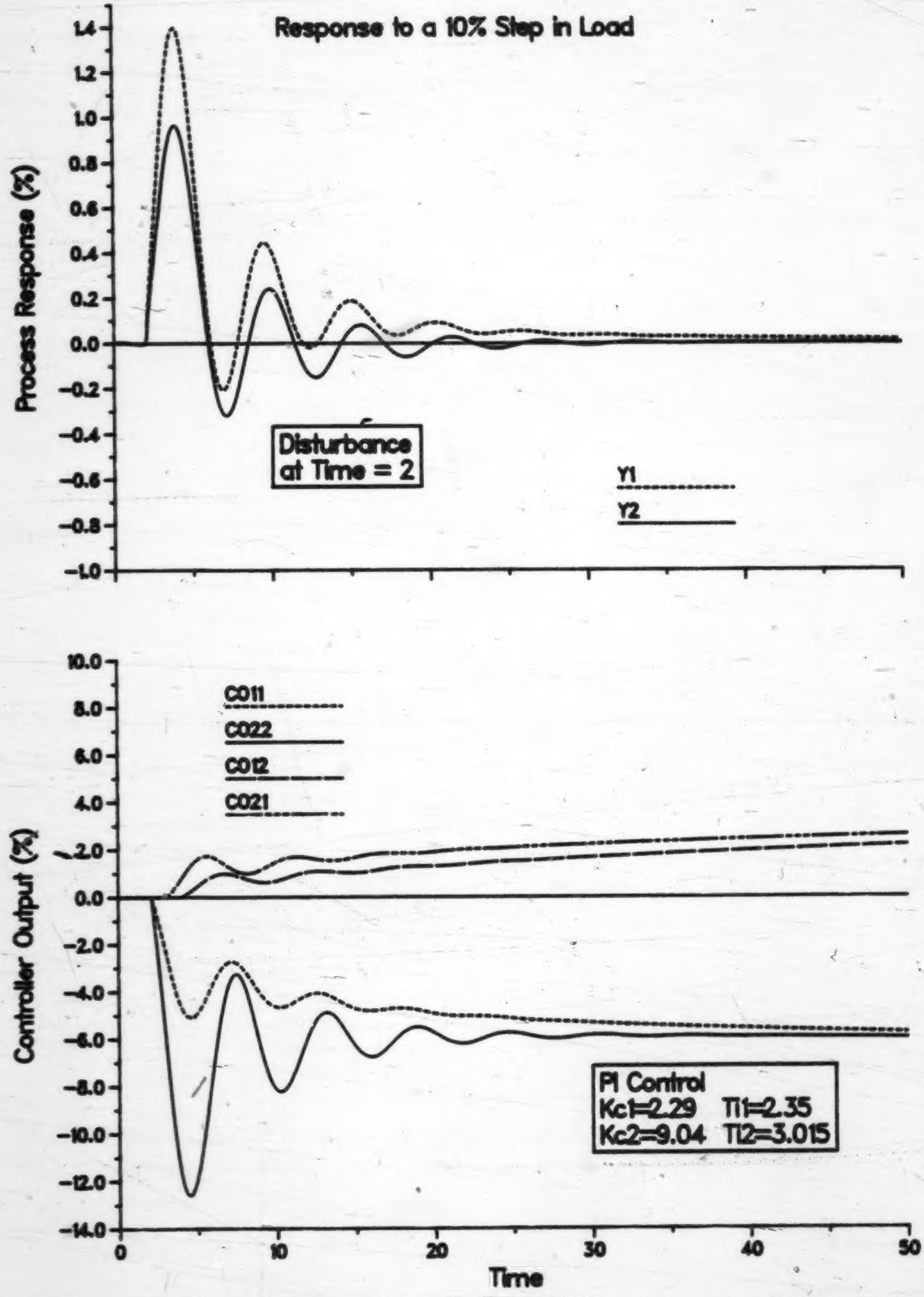


Figure E.10: Multiloop PI Control of the Original Process with Dynamic Decoupling, Load Disturbance Response

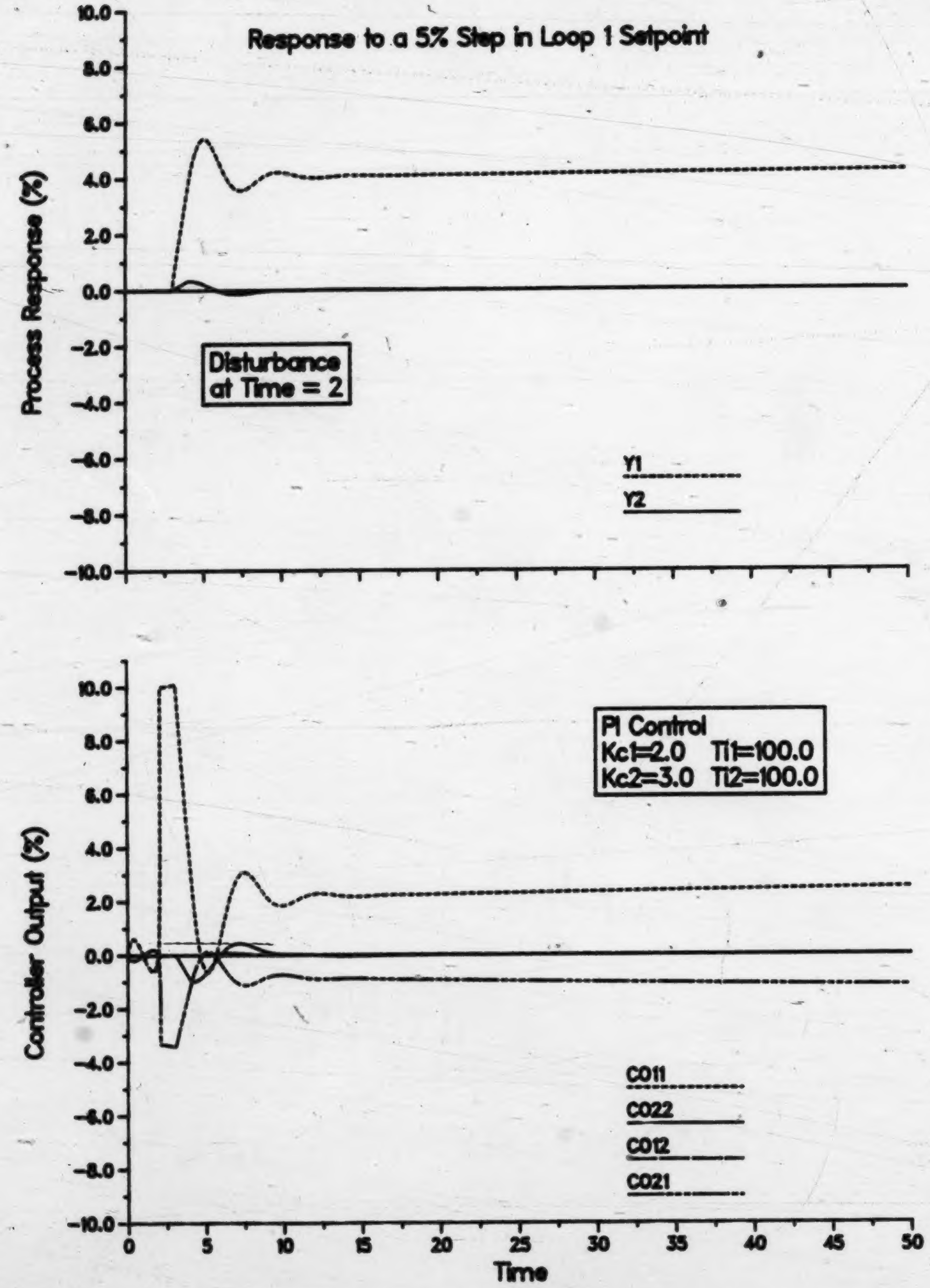


Figure E.11: Multiloop PI Control of a Process Requiring Predictive Dynamic Decoupling

decoupler. Clearly, the behavior of the control system has been degraded.

Another option in situations like this would be to only partially decouple. That is, the effect of changes in loop 1 could be eliminated from loop 2 by using G^*_{C12} and the second decoupling controller (G^*_{C21}) not used so changes in loop 2 would affect loop 1 as a load disturbance. This would effectively introduce a zero in the upper off-diagonal element of the closed loop transfer function matrix.

E.2.8 Decoupled Multi-loop PI Control with a Lead Time Dominant Decoupler

To complete this analysis of decoupling behavior, unlike for the previous examples which used a process model that resulted in lag time dominant decouplers, a lead time dominant decoupler was considered. Using to equation (E.7), a new plant transfer function matrix was constructed by interchanging the denominators of the terms in the second row to yield the new plant model:

$$G_p(s) = \begin{bmatrix} \frac{2e^{-1s}}{5s+1} & \frac{1e^{-3s}}{40s+1} \\ \frac{1e^{-2s}}{20s+1} & \frac{2e^{-1s}}{30s+1} \end{bmatrix} \qquad \text{(E.16)}$$

Clearly, G^*_{C12} is unchanged from that given by equation (E.13). However, the dynamics of the off-diagonal transfer function in row two are somewhat faster (i.e. has a smaller time constant) than the diagonal transfer function resulting in the decoupler:

$$G^{*}_{C21}=\frac{-0.5(30s+1)e^{-1s}}{20s+1} \tag{E.17}$$

which replaces that given in equation (E.14).

Two different simulations using the plant transfer function matrix in equation (E.16) were performed using the same PI control as before with the controller constants calculated as discussed earlier in this section to arrive at the controller constants:

$$K_{p1} = 2.29$$

$$\tau_{I1} = 2.35$$

and:

$$K_{p2} = 13.5$$

$$\tau_{I2} = 3.11$$

The closed-loop response, for a 5% step increase in the loop 1 setpoint, given in Figure E.12 is for the original plant before decoupling while Figure E.13 shows the response when decoupling is used. It can be seen from these figures that an offset in the loop 1 response of 0.5% exists after 100 minutes of simulation time, indicating poor system performance. This poor performance is due to the fact that the controller tuning constants had to be changed to $K_{C1} = K_{C2} = 2$ and $\tau_{I1} = \tau_{I2} = 100$, before a stable response could be obtained in either system.

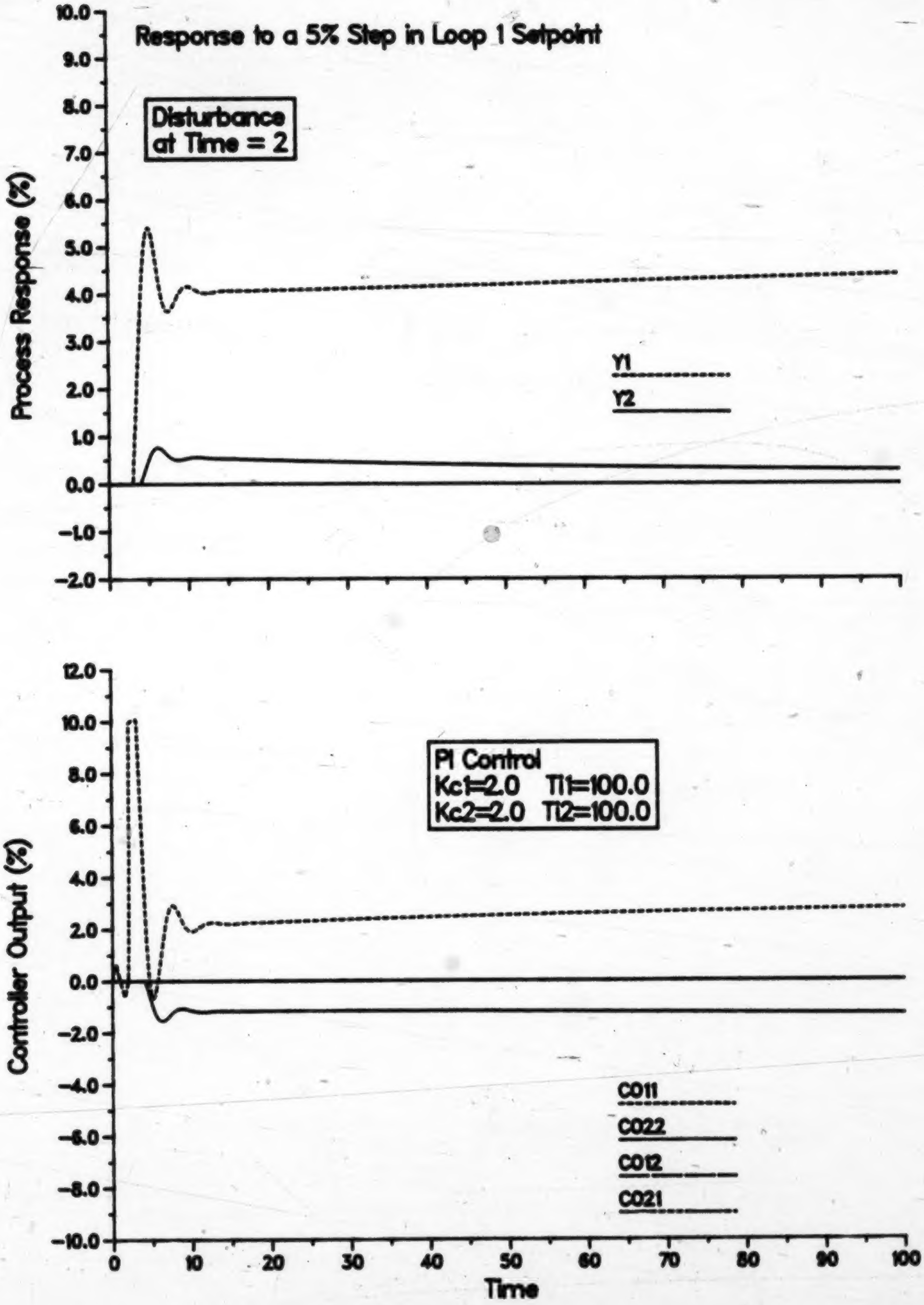


Figure E.12: Multiloop PI Control of a Process with Slow Loop 2 Direct Transmission Dynamics

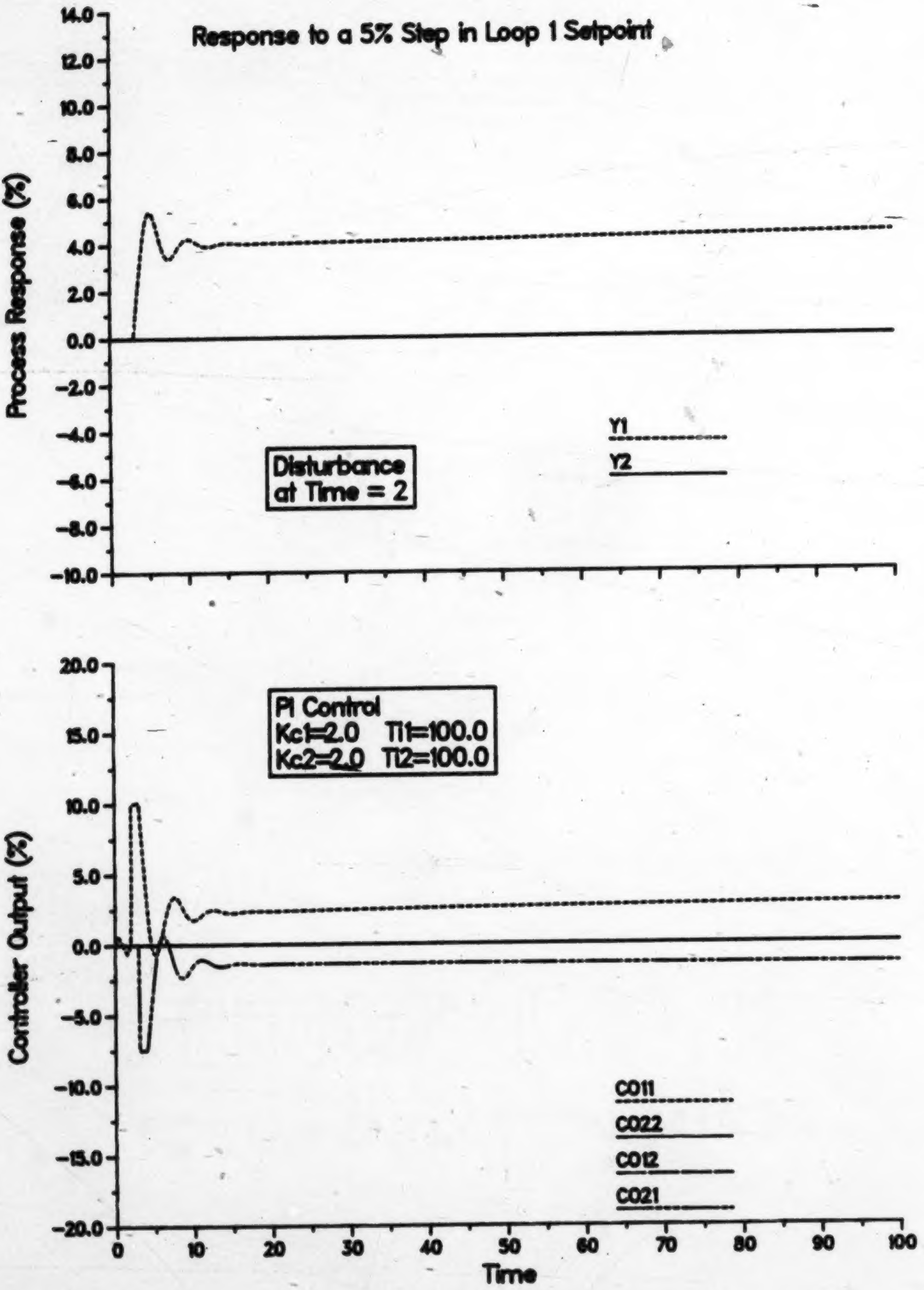


Figure E.13: Multiloop PI Control of a Process with Slow Loop 2 Direct Transmission Dynamics Using Dynamic Decoupling

F.1 Time Delay Compensator - Subroutine TDC

The FORTRAN implementation of the compensator described in section 2.6.2 is given in Figure D.15. The code in this figure, when reviewed in conjunction with the discussion in section 2.6.2, is easily understood. The subroutine performs all the calculations for the G_k block in Figure 2.21 for a process whose model consists of up to 25 first order plus time delay transfer functions. This routine is written to be compatible with DYFLO2 and is placed in the control subroutine group.

To verify that the routine functions correctly, an open loop simulation was performed for the block diagram given in Figure F.1. A step increase in the loop 1 input variable was used to excite the system.

The 2x2 process transfer function matrix used for this simulation was:

$$G_p(s) = \begin{bmatrix} \dfrac{2e^{-1s}}{5s+1} & \dfrac{1e^{-3s}}{40s+1} \\ \dfrac{1e^{-2s}}{30s+1} & \dfrac{2e^{-1s}}{20s+1} \end{bmatrix} \qquad (F.1)$$

This process was treated as a "black box" for the purposes of simulating the actual plant and was also used in the design of the time delay compensator. The same plant was used in Appendix E for the decoupling analysis (cf. equation

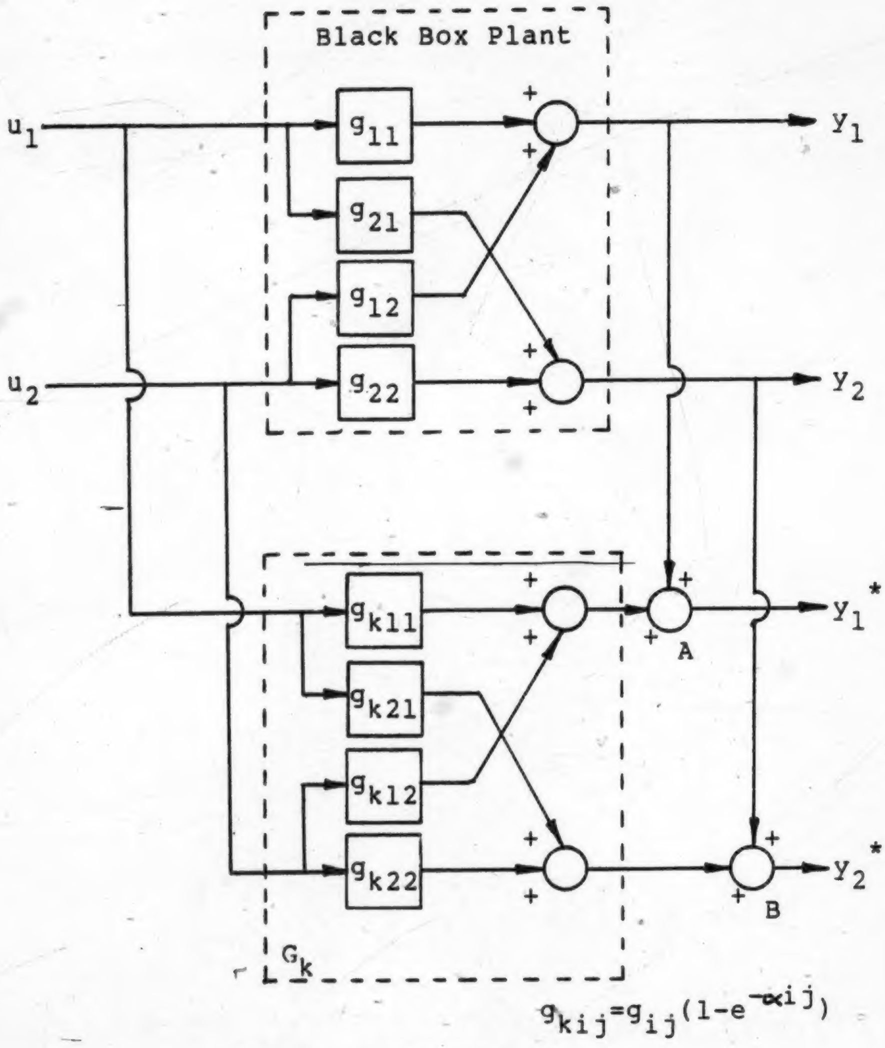

Black Box Plant
u_1
u_2
g_{11}
g_{21}
g_{12}
g_{22}
y_1
y_2
g_{k11}
g_{k21}
g_{k12}
g_{k22}
y_1^*
y_2^*
A
B
G_k
$g_{kij}=g_{ij}(1-e^{-\alpha ij})$

Figure F.1: Block Diagram for Time Delay Compensation Subroutine Verification

E.7).

The results of this simulation are presented in Figures F.2 and F.3. The solid line in both figures is the output of the time delay compensation routine plus the black box plant output while the dashed line is just the black box plant output. It can be seen that the combined output from the time delay compensator and the actual plant is simply the undelayed plant response, as expected. The DYFLO2 executive program used for this simulation is given in Figure F.4.

F.2 Time Delay Compensation

F.2.1 Introduction

To examine the behavior of the time delay compensator in various situations, several simulations under somewhat idealized conditions were performed. Again, the process model used for these simulations is that given as equation (F.1). Using this transfer function matrix allows for comparison of the results obtained in this appendix with those obtained for the decoupling analysis discussed in Appendix E. Also, the interaction analysis and controller tuning constant calculated estimates were discussed in Appendix E (sections E.2.2 and E.2.3 respectively), and so, need not be considered in this section.

Several different simulations were performed using various forms of the process model in equation (F.1). Each variation will be discussed in conjunction with the appropriate case study.

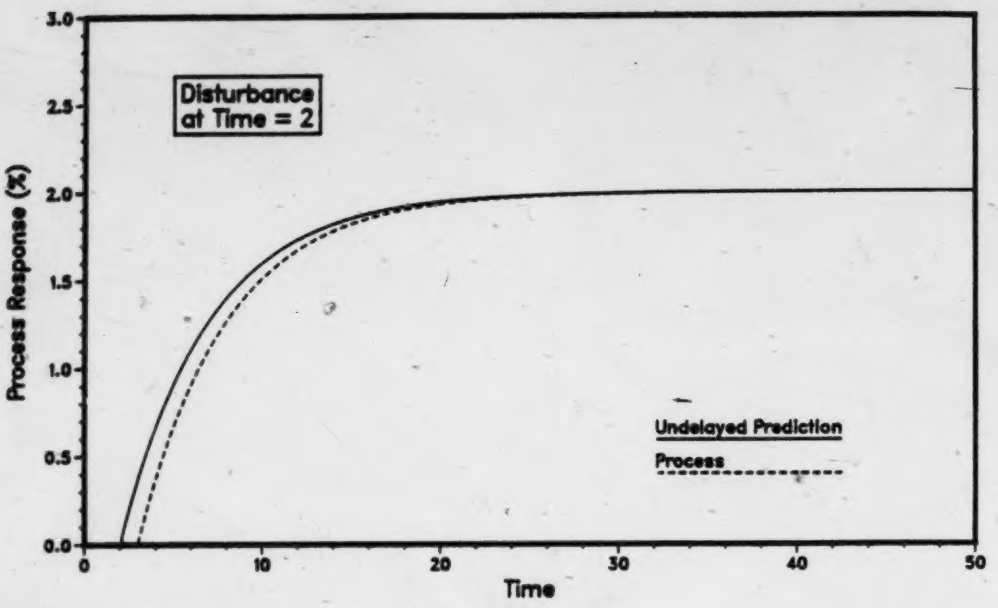


Figure F.2: Loop 1 Open Loop Response of the Time Delay Compensator to a Step Input

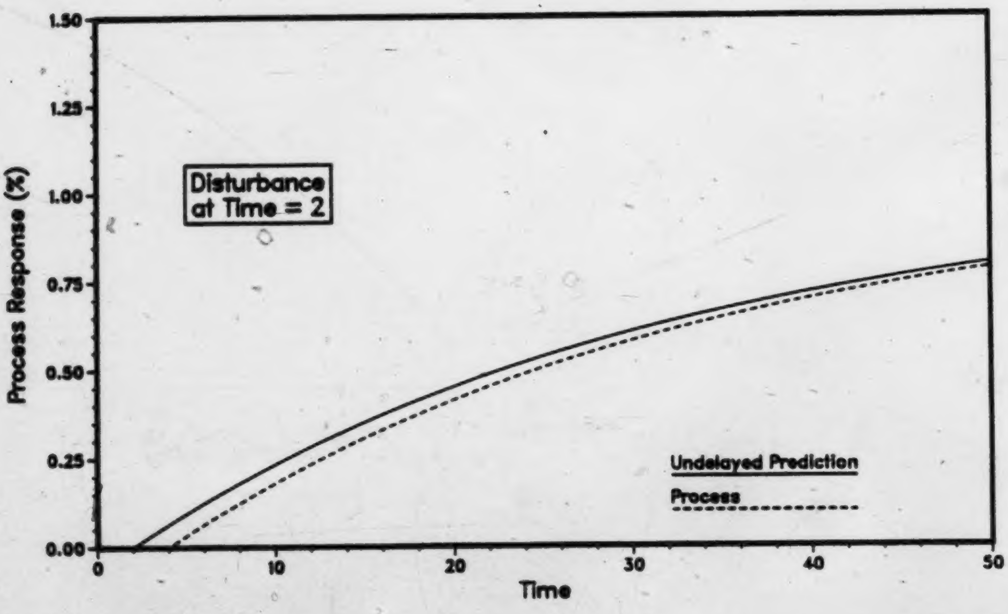


Figure F.3: Loop 2 Open Loop Response of the Time Delay Compensator to a Step Input

```
C
C   TIME DELAY COMPENSATOR (GENERALIZED SMITH PREDICTOR)
C   TEST DRIVER
C
      COMMON/CINT/T,DT,JS,JN,DXA(500),XA(500),IO,JS4,NC,NCON
      COMMON/SIN/ SIMT,RINT,PT,TIN
      REAL KP(2,2),KL(2),TAU(2,2),TAUL(2),DX(2),DY(2),DD(2),YA(25,500)
      REAL KC1,KC2,KLL1,KLL2,DIN(2),DOUT(2)
      INTEGER NTL(2,2),NTLL(2)
      COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
      LOGICAL NF
C
C   INITIALIZE SYSTEM VARIABLES
C
      NF=.FALSE.
      TIN=0.
      RINT=0.1
      SIMT=50.
      PT=1.
      DISTIN=1.99
      IO=1
      JN=0
      JS=0
      JS4=0
C
C   INITIALIZE LINEAR MODEL PARAMETERS
C
      CALL LMOD(RINT)
      CO11=50.
      CO22=50.
      DCO11=0.
      DCO22=0.
      CO12=0.
      CO21=0.
      KLL1=-KP(1,2)/KP(1,1)
      KLL2=-KP(2,1)/KP(2,2)
      SP1=50.
      SP2=50.
      KC1=2.29
      KC2=9.04
      TI1=2.35
      TI2=3.015
C      KC1=2.
C      KC2=2.
C      TI1=100.
C      TI2=100.
      PB1=100./KC1
      PB2=100./KC2
      RPT1=1./TI1
      RPT2=1./TI2
      ZR1=0.
      ZR2=0.
      FSRDG1=100.
      FSRDG2=100.
      AXN1=-1.
      AXN2=-1.
      OI1=50.
      OI2=50.
      SSY1=50.
      SSY2=50.
      Y1=SSY1
      Y2=SSY2
      Y1STAR=0.
      Y2STAR=0.
      DLOAD=0.
      TD1=RINT*FLOAT(NTL(1,2)-NTL(1,1))
      TD2=RINT*FLOAT(NTL(2,1)-NTL(2,2))
C       TD1=0.
C       TD2=0.
      IF(TD1 .LE. 0.) TD1=0.
      IF(TD2 .LE. 0.) TD2=0.
      WRITE(6,1)
    1 FORMAT(/,5X,'LEAD - LAG COMPENSATOR TRANSFER FUNCTIONS',//)
      WRITE(6,2) TAU(1,1),TAU(2,2)
    2 FORMAT(19X,'(',F7.3,'s+1)',32X,'(',F7.3,'s+1)')
      WRITE(6,3) KLL1,TD1,KLL2,TD2
    3 FORMAT(7X,'G12(s)=',F5.2,12('-'),'*exp(-',F5.2,'s)',
     +        7X,'G21(s)=',F5.2,12('-'),'*exp(-',F5.2,'s)')
      WRITE(6,4) TAU(1,2),TAU(2,1)
    4 FORMAT(20X,F7.3,'s+1',34X,F7.3,'s+1',//)
      WRITE(6,5)
    5 FORMAT(4X,'TIME',7X,'DCO11',7X,'Y1STAR',5X,' CO12   ',4X,
     +'OUTPUT 1',6X,'DCO22',7X,'Y2STAR',5X,' CO21   ',4X,'OUTPUT 2',
     + 6X,'LOAD',/)
C
C   MAIN SIMULATION LOOP
C
C   DERIVATIVE SECTION
C
C   DISTURBANCE
C
   10 IF(TIN .GE. DISTIN) SP1=55.
       DD(1)=DLOAD
       DD(2)=DLOAD
C
C   ADJUST PRINT INTERVAL
C
        IF(TIN .GE. DISTIN) PT=.1
        IF(TIN .GE. (DISTIN*5.)) PT=1.
C
C   DERIVATIVE FOR INTEGRAL OF THE SQUARE OF THE ERROR CALC.
C   NOTE: THIS CALC. IS BASED ON THE BEST POSSIBLE RESPONSE
C         EXPECTED FROM THE SYSTEM WHEN TIME DELAYS ARE
C         CONSIDERED
C
        IF(TIN .GE. (DISTIN+TAU(1,1))) DERR1=ABS(SP1-Y1)**2.
        IF(TIN .GE. (DISTIN+TAU(2,2))) DERR2=ABS(SP2-Y2)**2.
C
C   CONTROLLER INPUT CALCULATIONS
C
        CI1=Y1+Y1STAR
        CI2=Y2+Y2STAR
C
C   CALCULATE CONTROLLER OUTPUT DEVIATION VARIABLES
C
        DCO11=CO11-SSY1
        DCO22=CO22-SSY2
C
C   PLANT INPUTS
C
```

Figure F.4: Time Delay Compensation DYFLO2 Program Source Code Listing

```
      DX(1)=DCO11+CO12
      DX(2)=DCO22+CO21
C
C   TEST FOR PRINT AND FINISH
C
      CALL PRNTF(PT,SINT,NF,TIN,DCO11,Y1STAR,CO12,DY(1),DCO22,
     +     Y2STAR,CO21,DY(2),DLOAD)
      IF(NF) GOTO 20
C
C   INTEGRATION SECTION
C
      CALL INT1(TIN,RINT,IO)
C
C   CONTROLLER CALCULATIONS.
C
      CALL CONTR2(CI1,CO11,ZR1,FSRDG1,SP1,AXN1,PB1,RPT1,OI1)
      IF(ABS(CO11) .LT. 1.0E-10) CO11=0.
      CALL CONTR2(CI2,CO22,ZR2,FSRDG2,SP2,AXN2,PB2,RPT2,OI2)
      IF(ABS(CO22) .LT. 1.0E-10) CO22=0.
C
C   DECOUPLE
C
      CALL LLCOMP(CO12,DCO22,TAU(1,1),TAU(1,2),TD1,KLL1,1)
      CALL LLCOMP(CO21,DCO11,TAU(2,2),TAU(2,1),TD2,KLL2,2)
C
C   SET DECOUPLER OUTPUT TO 0 IF NO DECOUPLING
C     (TURN ON OR OFF WITH COMMENT CARDS)
C
C     CO12=0.
C     CO21=0.
C
C   DELAY COMPENSATION
C
      M=2
      N=2
      DIN(1)=DX(1)
      DIN(2)=DX(2)
      CALL TDC(DOUT,DIN,KP,TAU,NTL,M,N)
      Y1STAR=DOUT(1)
      Y2STAR=DOUT(2)
C
C   PLANT CALCULATIONS
C
      CALL I2BY2(TIN,RINT,1)
C
C   PLANT OUTPUT CALCULATIONS
C
      Y1=DY(1)+SSY1
      Y2=DY(2)+SSY2
C
C   PERFORMANCE CALCULATIONS
C
      CALL INTG(ERR1,DERR1)
      CALL INTG(ERR2,DERR2)
      GOTO 10
C
C   TERMINATION SECTION
C
   20 WRITE(6,6) ERR1,ERR2
    6 FORMAT(/,5X,'ISE FOR LOOP 1 = ',F12.6,/,5X,
     +       'ISE FOR LOOP 2 = ',F12.6)
      STOP
      END
C
C-------------------------------------------------------------------
C
C   2x2 COUPLED SYSTEM PARAMETER INITIALIZATION ROUTINE
C
C-------------------------------------------------------------------
C
      SUBROUTINE LMOD(RINT)
      REAL KP(2,2),KL(2),TAU(2,2),TAUL(2),DX(2),DY(2),DD(2),YA(25,500)
      INTEGER NTL(2,2),NTLL(2)
      COMMON/IN2BY2/KP,KL,TAU,TAUL,NTL,NTLL,DX,DY,DD
C
C   INITIALIZE TRANSFER FUNCTION MODEL PARAMETERS
C
C---> GAINS
C
      KP(1,1)=2.
      KP(2,1)=1.
      KP(1,2)=1.
      KP(2,2)=2.
      KL(1)=1.
      KL(2)=1.
C
C---> TIME CONSTANTS
C
      TAU(1,1)=5.
      TAU(2,1)=30.
      TAU(1,2)=40.
      TAU(2,2)=20.
      TAUL(1)=10.
      TAUL(2)=15.
C
C---> DEAD TIME
C
      TD11=1.
      TD21=2.
      TD12=3.
      TD22=1.
      TDLD1=0.
      TDLD2=0.
C
C---> S.S. OPERATING CONDITIONS
C
      DX(1)=0.
      DX(2)=0.
      DY(1)=0.
      DY(2)=0.
C
C   CALCULATE NUMBER OF INTEGRATION STEPS IN TIME DELAYS
C
      NTL(1,1)=INT(TD11/RINT)
      NTL(2,1)=INT(TD21/RINT)
      NTL(1,2)=INT(TD12/RINT)
      NTL(2,2)=INT(TD22/RINT)
      NTLL(1)=INT(TDLD1/RINT)
      NTLL(2)=INT(TDLD2/RINT)
      RETURN
      END
End of file
```

Figure F.4 continued

F.2.2 Integral of the Square of the Error Calculation Technique

In order to obtain a more quantitative measure of the control system performance, an integral of the square of the error (ISE) calculation was included in every simulation in this appendix. The choice of this particular performance index from the many available was arbitrary.

This index is based on the difference between desired and actual outputs. This error is squared and then integrated over time as follows:

$$ISE = \int_0^t \epsilon(t)^2 dt \qquad (F.2)$$

where the error is defined by the equation:

$$\epsilon(t) = y_{sp}(t) - y(t) \qquad (F.3)$$

Squaring the error has the effect of weighting large errors much more heavily than small errors and also prevents negative and positive errors from cancelling each other.

One disadvantage that immediately becomes apparent is that the sensitivity of this index is lost when process time delays are present. This can be exemplified by considering the hypothetical system response for a unit step increase in setpoint shown in Figure F.5. The process has a time delay associated with it and the best possible response that can be expected is shown in the figure by the dashed line. Clearly, the error represented by the shaded block between the setpoint curve and the best response curve is

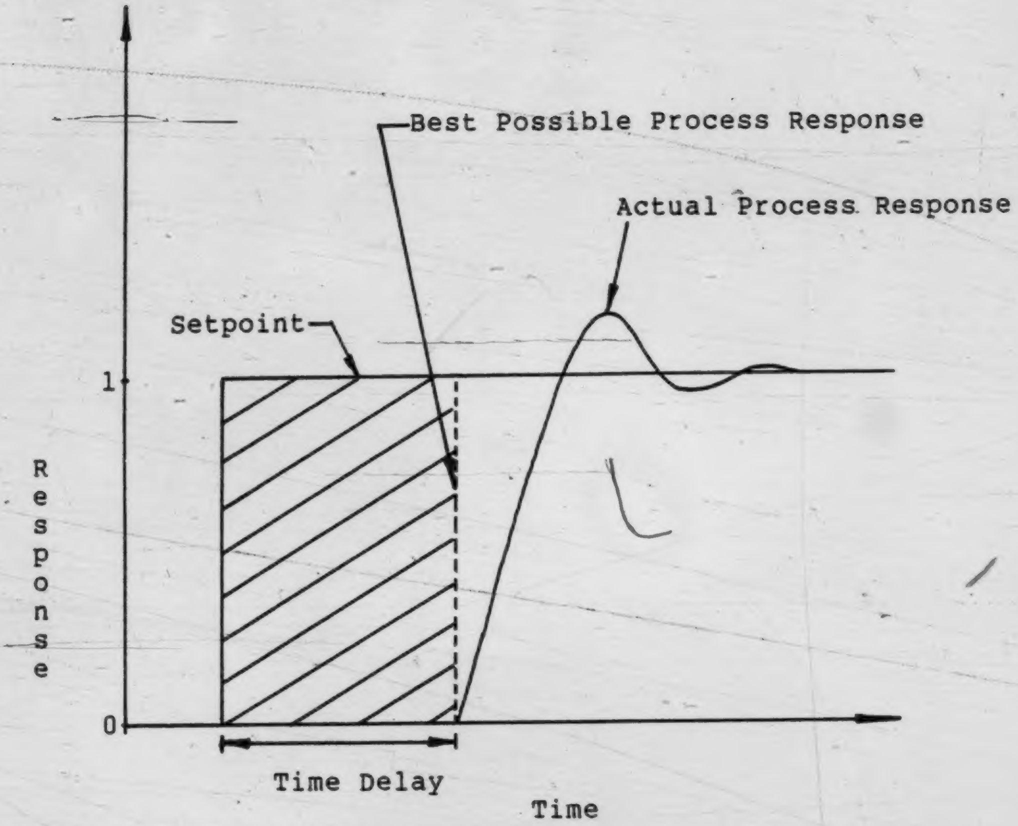

Best Possible Process Response
Actual Process Response
Setpoint
1
0
Response
Time Delay
Time

Figure F.5: Demonstration of the Decrease in ISE Sensitivity When Process Time Delays Are Present

very large and has a tendency to bias the ISE performance index. This problem can be alleviated by using the best possible response in the error calculation instead of the setpoint value as follows:

$$\epsilon(t) = y_{best}(t) - y(t) \qquad (F.4)$$

This eliminates the large unavoidable error from the ISE calculation and improves the sensitivity of the index thus allowing for more subtle differences in control system performance to be detected. Equation (F.4) was used with equation (F.2) for all the ISE calculations in this appendix.

F.2.3 Multi-loop PI Control of the Original System

This first case is a repeat of the first case considered in section E.2.4 and is presented for convenience. Proportional plus integral (PI) controllers with constants as calculated in Appendix E were used in the main loops for all simulations. Unless otherwise noted, the controller constants used for all simulations discussed in this section were:

$$K_{p1} = 2.29$$

$$K_{p2} = 9.04$$

and:

$$T_{I1} = 2.35$$

$$T_{I2} = 3.02$$

Figure F.6 shows the response of the original, uncompensated, system for a 5% step increase in the setpoint in loop 1. As a baseline measure of the control system performance, the integral of the square of the error (ISE) was calculated to be 41.8 and 5.97 for loop 1 and loop 2 respectively.

F.2.4 Time Delay Compensated Multi-loop PI Control of the Original System

Incorporating the Ogunnaike and Ray (1979) time delay compensator, as shown in the block diagram in Figure 2.22,

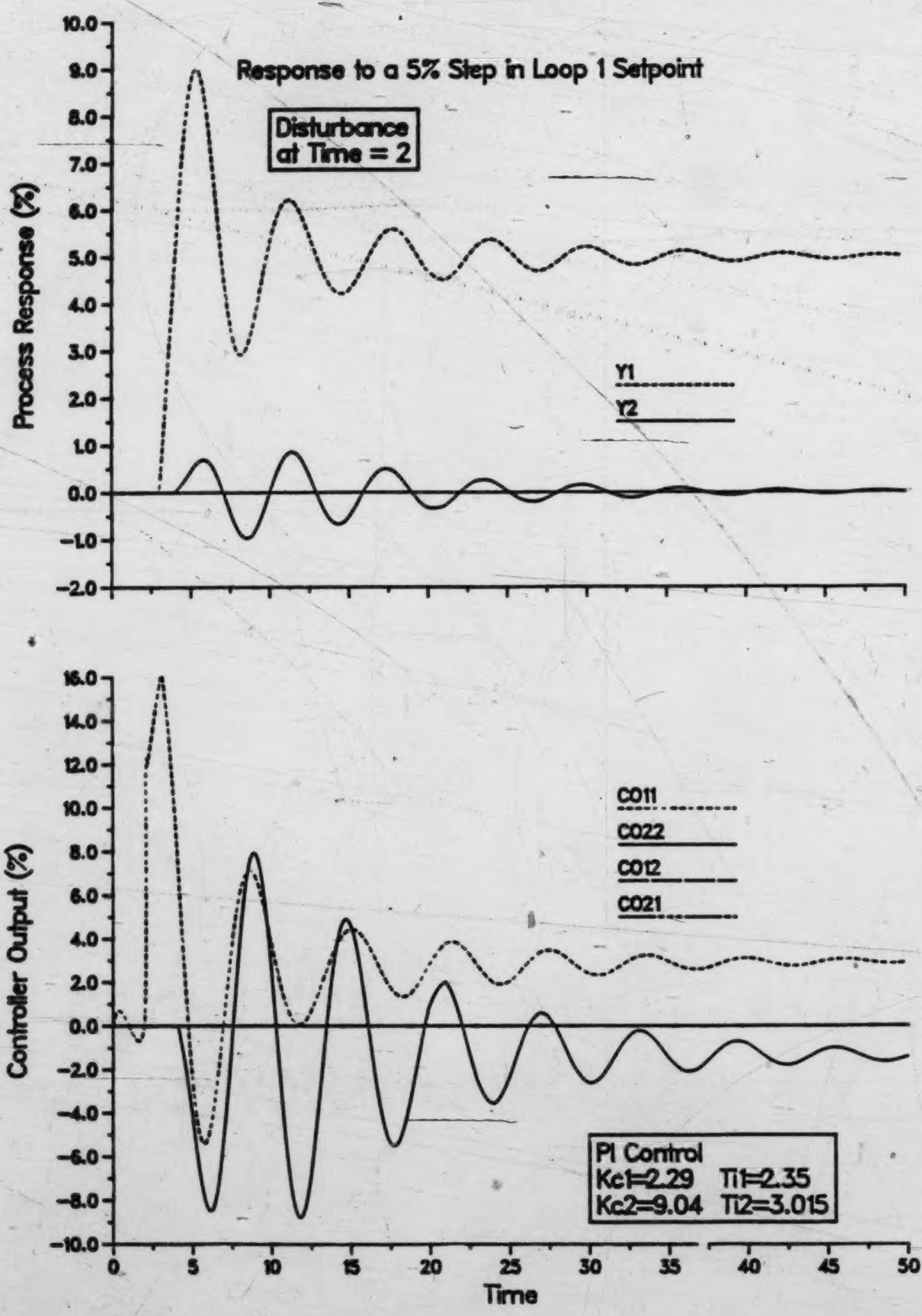


Figure F.6: Multiloop PI Control of the Original Process

improves the system performance immensely as can be seen from the responses in Figure F.7. The system response for this case resulted in ISE values of 0.377 for loop 1 and 0.0241 for loop 2. Comparison of these responses with those in Figure F.6 or the performance measures, clearly shows the benefits of time delay compensation.

F.2.5 Multi-loop PI Control with Faster Off-Diagonal Dynamics in the Process Model

A "stiffer" test of the benefits of time delay compensation occurs when an off (row) diagonal transfer function in the process transfer function matrix has a shorter time delay than the associated diagonal transfer function. This situation corresponds to the unrealizable decoupling controller case discussed in section E.2.7. The process model used for both the "black box" plant model and the compensator design was the same as given in equation (F.1), except the delay terms in the second row were interchanged.

The response of this system for a 5% step increase in the setpoint in loop 1 without time delay compensation is presented in Figure F.8. The ISE values for these responses were 33.4 and 3.08 for loop 1 and loop 2 respectively. It should be noted however, that the controller constants had to be detuned (K_{c1}, K_{c1} = 2; T_{I1}, T_{I2} = 100) before a stable response could be obtained.

F.2.6 Time Delay Compensated Multi-loop PI Control with Faster Off-Diagonal Dynamics in the Process Model

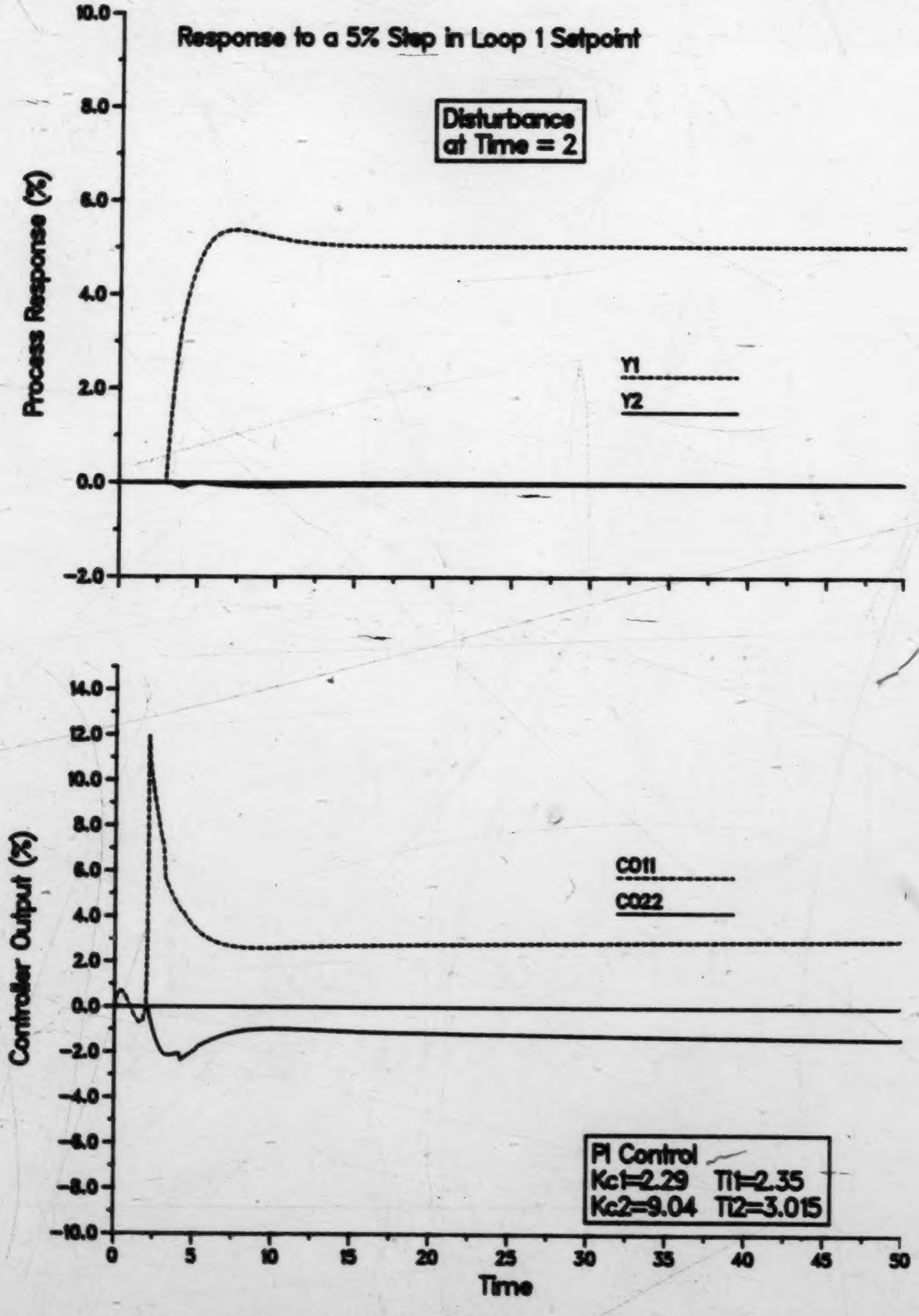


Figure F.7: Multiloop PI Control of the Time Delay Compensated Process

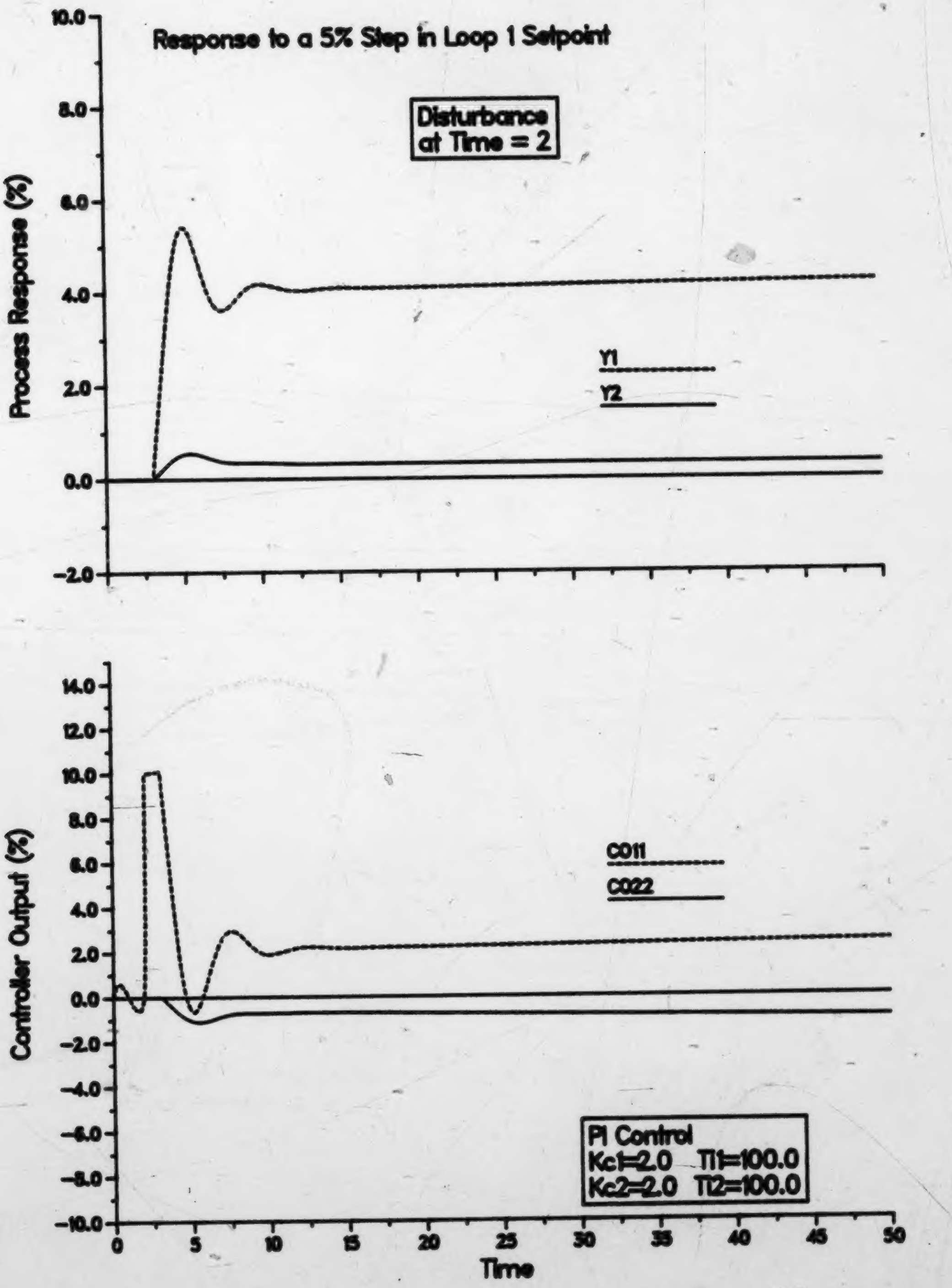


Figure F.8: Multiloop PI Control of a Process with a Short Off-diagonal Time Delay Element

Based on the results from sections F.2.3 and F.2.4, it is obvious that the situation described above can be improved on by using time delay compensation in the control system. Figure F.9 shows the response of the compensated system using the same disturbance with the original controller constants. In this case, the ISE for loop 1 was 0.371 while for loop 2 the ISE was 0.0341 which represents an improvement in the performance of about 99% for both loops! It should be noted that for use of decoupling there was no improvement in the performance of loop 1, but the effect of the disturbance in loop 2 was almost completely eliminated (ISE loop 1 = 35.0; ISE loop 2 = 2.3×10^{-5}). However, for this latter case, the PI controllers had to be detuned, as was the case for the completely uncompensated case (Figure F.8) to obtain stable system behavior.

F.2.7 Time Delay Compensated, Decoupled Multi-loop PI Control

Presumably it would be advantageous to combine time delay compensation and decoupling so that the benefits of both can be obtained within a given control system. A block diagram for a two by two interacting system is shown in Figure F.10. For convenience, the decoupling controller is shown as four blocks in this diagram to maintain its generality for all decoupling techniques. Although Ogunnaike and Ray considered only static decoupling in their original work, it is felt that dynamic decoupling has

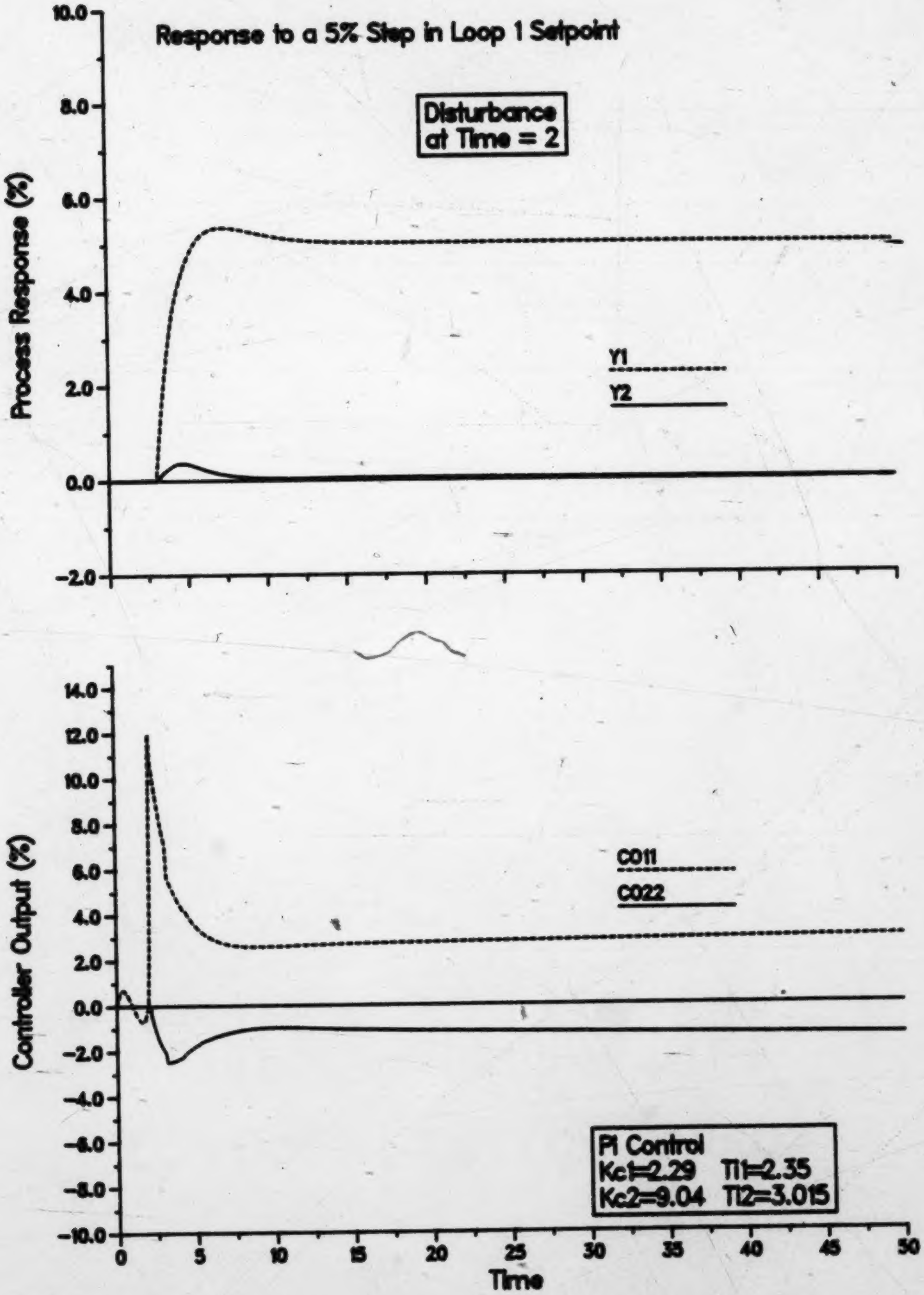


Figure F.9: Multiloop PI Control of a Process with a Short Off-diagonal Time Delay Element Using Time Delay Compensation

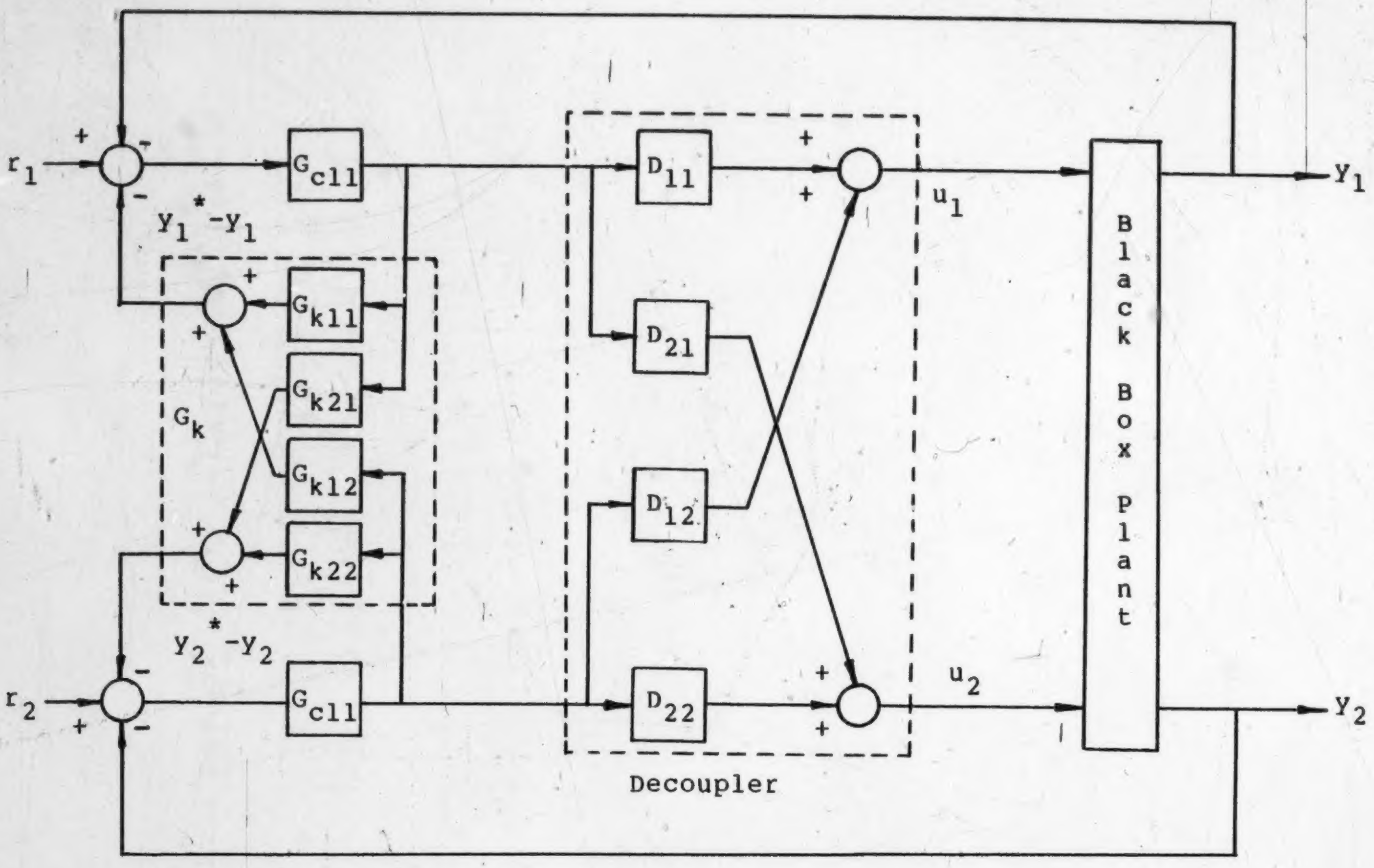

r_1
+
-
-
G_{c11}
$y_1^* - y_1$
G_k
G_{k11}
G_{k21}
G_{k12}
G_{k22}
+
+
+
+
$y_2^* - y_2$
r_2
+
-
-
G_{c11}
D_{11}
D_{21}
D_{12}
D_{22}
+
+
+
+
Decoupler
u_1
u_2
Black Box Plant
y_1
y_2

Figure F.10: Block Diagram for Decoupling and Time Delay Compensation Simulation

several advantages based on the results for these two techniques presented in Appendix E. This discussion will deal only with dynamic decoupling for this reason. It is interesting to note that, because of the nature of Ogunnaike and Ray's time delay compensation method, the controllers (G_{C11} and G_{C22}) are ideally operating on an undelayed model of the process. This leads one to think that the decoupling controllers should be designed for the undelayed model as well. However, a closer examination of the block diagram in Figure F.10 shows that this is not correct because the decoupling controllers are operating on the actual process. The decoupling technique used in this investigation follows the approach suggested by Zalkind (1967). Thus, the decoupler blocks on the two direct transmission paths (D_{11} and D_{22}) in Figure F.10 become 1. Using equation F.1 and equations (2.80) and (2.81), it follows that the decoupling controllers are:

$$D_{12} = G^{*}_{C12} = \frac{-0.5(5s+1)e^{-2s}}{40s+1} \qquad (F.5)$$

and:

$$D_{21} = G^{*}_{C21} = \frac{-0.5(20s+1)e^{-1s}}{30s+1} \qquad (F.6)$$

Figures F.11 and F.12 show the results of a simulation using the above decoupling controllers in conjunction with time delay compensation as the traces labelled Y1 and Y2, for loops 1 and 2, respectively. Included on these two figures are the responses obtained from simulations using only time delay compensation (cf. Figure F.7) and only

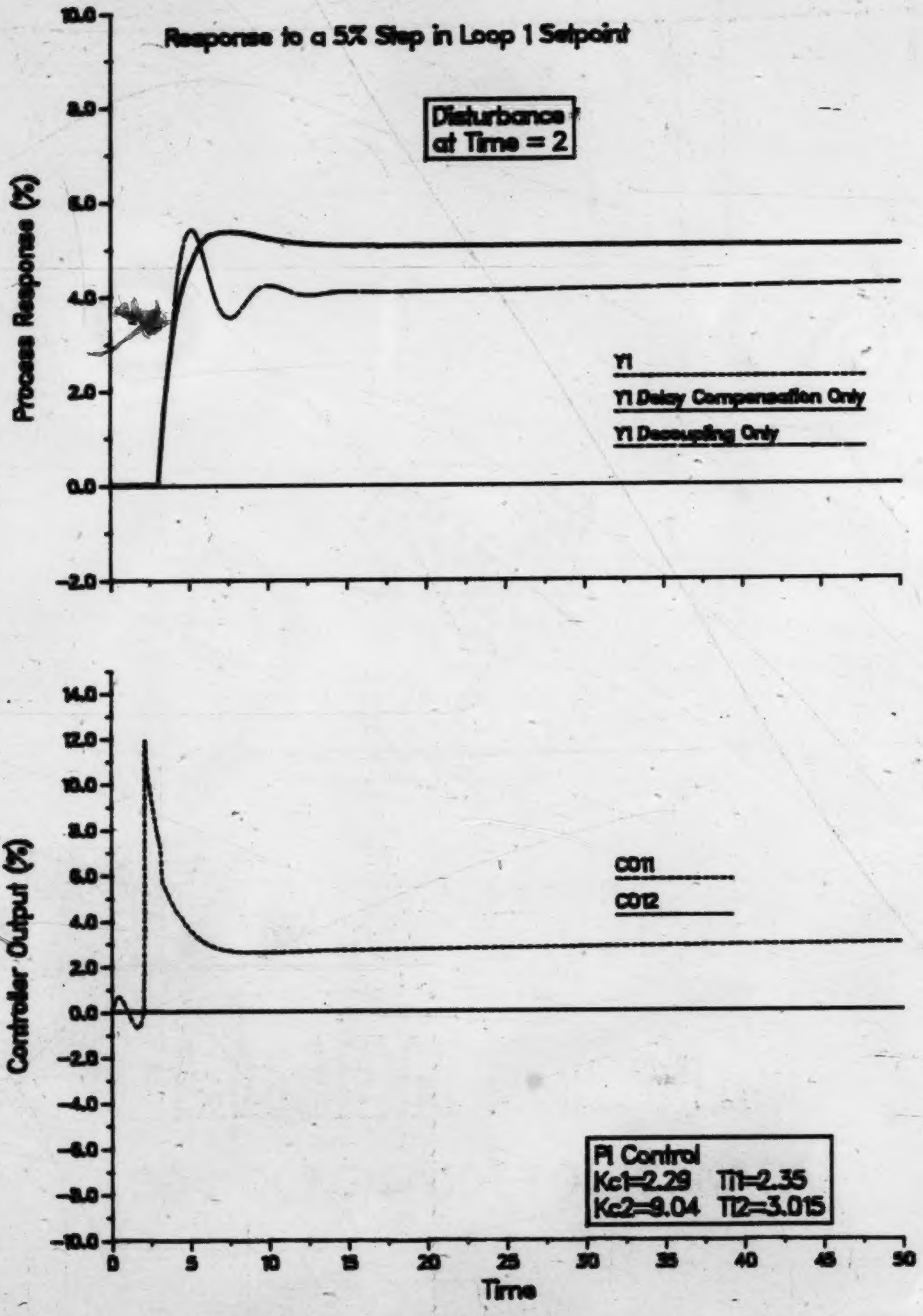


Figure F.11: Multiloop PI Control of the Original Process with Decoupling and Time Delay Compensation for Loop 1

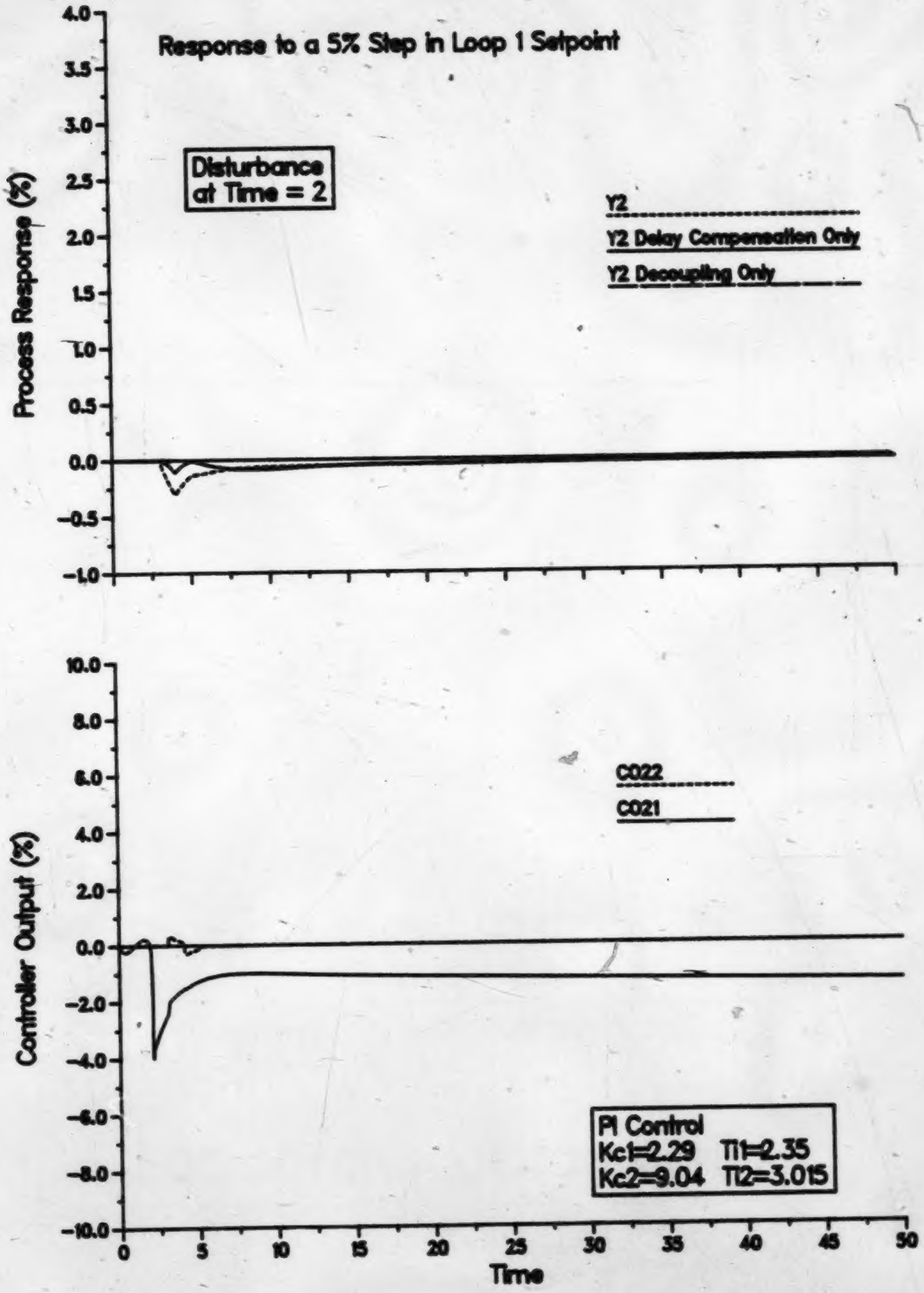


Figure F.12: Multiloop PI Control of the Original Process with Decoupling and Time Delay Compensation for Loop 2

decoupling (cf. Figure E.8). The fact that the time delay compensated and time delay compensated with decoupling responses in Figure F.11 overlay each other indicates that no improvement at all is realized in loop 1. Figure F.12 shows that the combination of the two compensation techniques actually causes a deterioration in the performance of loop 2 over either of the individually decoupled or time delay compensated cases. Similar behavior was observed by Ogunnaile and Ray (1979) using a steady state decoupling technique.

The explanation for this behavior follows from a mathematical analysis of the block diagram in Figure F.10. An alternate representation of this block diagram is shown in Figure F.13 and proves to be more convenient for the analysis. Ignoring the measurable disturbances, the closed loop transfer function, in matrix notation, is:

$$y = [I + G_p D G^*_c]^{-1} G_p D G^*_c r \qquad \text{(F.7)}$$

which is similar to equation 2.95 except for the presence of the decoupling matrix, D. Following the discussion in section 2.6, the closed loop transfer function can be written as:

$$y = G_p D [I + G_c G^*_p + G_c G_p [D - I]]^{-1} G_c r \qquad \text{(F.8)}$$

Comparing this equation with equation (2.100), it can be seen that, because of the term containing G_p, the characteristic polynomial still contains time delays. This is true regardless of the form of decoupling used. Clearly this defeats one of the main purposes of time delay

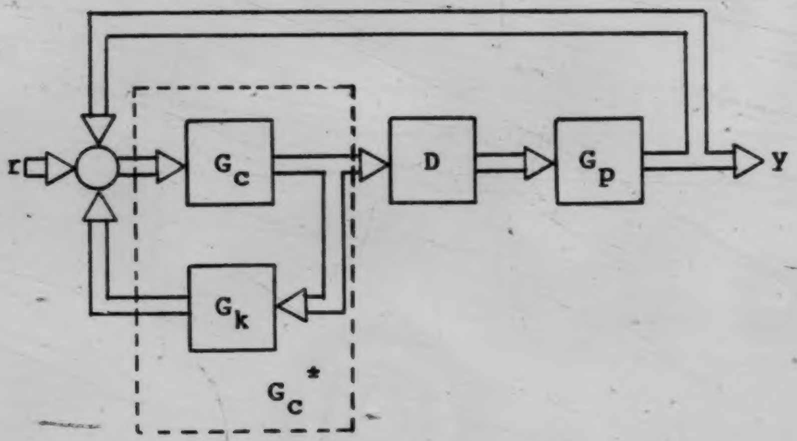


Figure F.13: Multivariable Block Diagram Representation for Decoupling and Time Delay Compensation Transfer Function Analysis

compensation which is to remove delays from the characteristic equation. Furthermore, it can be shown (after a great deal of algebra) that equation (F.8) is still coupled through the undelayed process model, G^*_p, so inclusion of the time delay compensator defeats the decoupling action. The fact that the two compensation techniques essentially fight each other is again born out by the ISE measure. The performance measure indicates that the behavior of loop 2 has degraded slightly; ISE = 0.0290 versus 0.0241 for the time delay compensation only case and ISE = 0.0 for the decoupling only case. Only a marginal improvement in the performance of loop 1 was noted over the time delay compensation case; ISE = 0.344, versus 0.377 for use of only time delay compensation. This suggests that Ogunnaike and Ray's time delay compensation technique should not be employed in conjunction with decoupling in the design of a control system.

A possible exception to this recommendation may arise when the process model is such that a predictive decoupler results (see sections E.2.7 and F.2.4). As mentioned earlier, substantial improvements were gained when either time delay compensation or decoupling techniques were employed in this situation. Figure F.14 gives the response of the system when both decoupling and time delay compensation were used. Although improvements over the response in Figure F.9 are not readily apparent, the ISE performance measure has decreased slightly in both loops (ISE loop 1 = 0.349 versus 0.371 for only time delay compensation and ISE loop 2 = 0.0287 versus 0.0341). It should be noted that the PI controllers did not need to be detuned to obtain a stable system response as was the case for use of only decoupling.

F.2.8 Time Delay Compensated, Decoupled Multi-loop PI Control - An Alternate Approach

An alternate approach for the design of a control system incorporating both time delay compensation and decoupling controllers is to decouple the plant and then use SISO Smith predictors in each seperate loop. This scheme is shown in

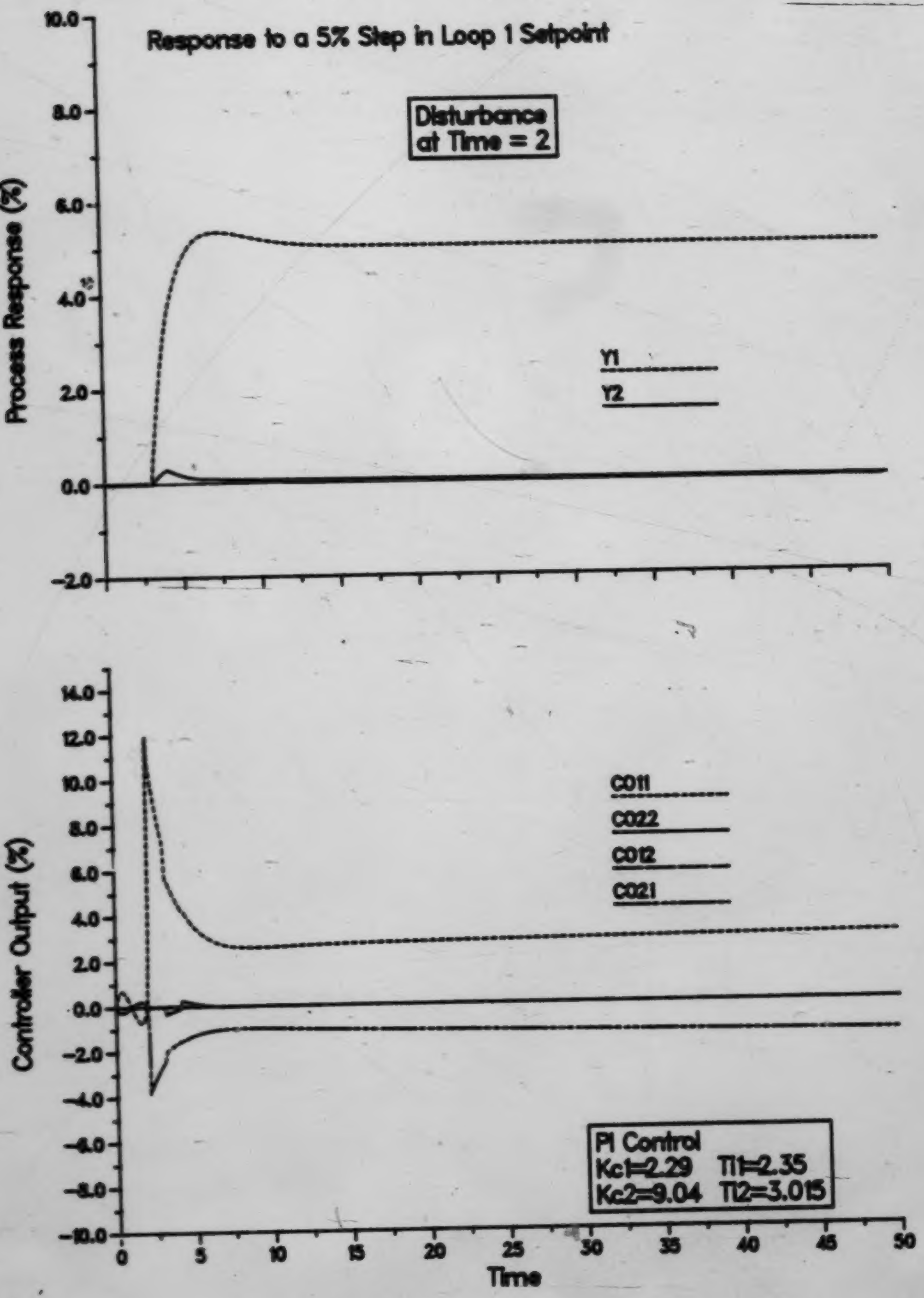


Figure F.14: Multiloop PI Control of a Process with a Short Off-diagonal Time Delay Element Using Decoupling and Time Delay Compensation

the block diagram form in Figure F.15. In this case the individual Smith predictors are designed for the diagonalized process transfer function matrix model that results from decoupling as shown in the dotted box labeled G'_p in the figure. Using Zalkind's noninteracting controller design technique, the decoupler is:

$$D = \begin{bmatrix} 1 & -G_{12}/G_{11} \\ -G_{21}/G_{22} & 1 \end{bmatrix} \qquad (F.9)$$

and so, the appropriate process model for use in the time delay compensator design is:

$$G'_p = \begin{bmatrix} (G_{11}G_{22}-G_{12}G_{21})/G_{22} & 0 \\ 0 & (G_{22}G_{11}-G_{21}G_{12})/G_{11} \end{bmatrix} \qquad (F.10)$$

As can be seen, this approach involves combination of the original process transfer functions so the resulting time delay compensator is much more complex than that designed using Ogunnaike and Ray's method. Additional analysis is beyond the scope of this work and the reader should consult the thesis of Vermeer (1986).

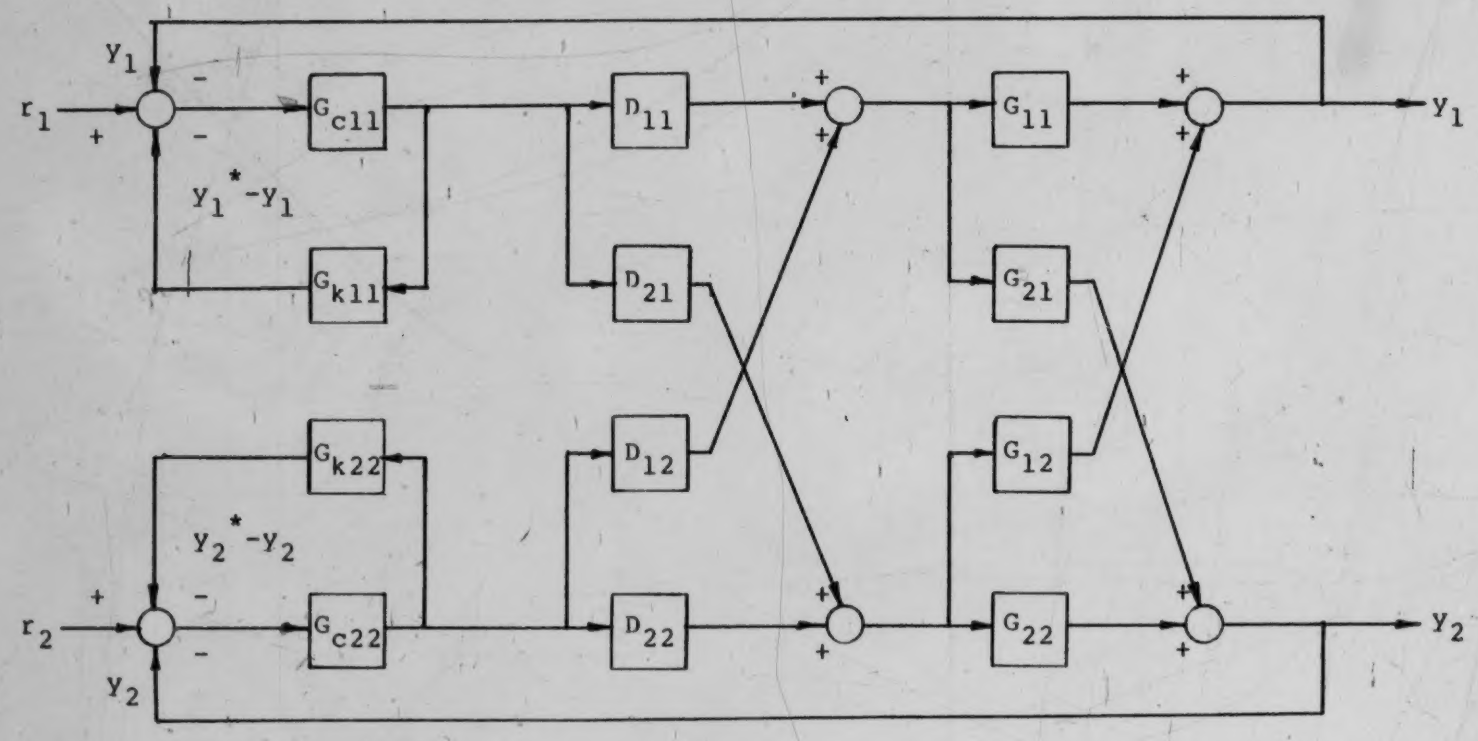

y_1
r_1
+
−
−
$y_1^* - y_1$
G_{c11}
G_{k11}
G_{k22}
G_{c22}
$y_2^* - y_2$
+
−
−
r_2
y_2
D_{11}
D_{21}
D_{12}
D_{22}
G_{11}
G_{21}
G_{12}
G_{22}
+
+
y_1
y_2

Figure F.15: Block Diagram for Alternate Decoupling with Time Delay Compensation Control System

Appendix G: Sample Terminal Sessions

G.1 Lake Dufault Grinding Circuit Simulator Sample Session

```
#SO GMCD:MINSIM.2.EXE

#MINSIM2
 "-MIN.OUT" does not exist.
 "-STC.OUT" does not exist.
 "-STC.PAR1" does not exist.
 "-STC.PAR2" does not exist.
 "-STC.SET" does not exist.



                     CLOSED CIRCUIT GRINDING SIMULATOR
                                 COPYRIGHT
                            UNIVERSITY OF ALBERTA
                     DEPARTMENT OF CHEMICAL ENGINEERING
                         WRITTEN BY: G.B. MCDOUGALL


                          INTERACTIVE VERSION 2.7
                                 86/03/25



         Do you require instructions? (Y/N)

Y


    This program is menu driven and requires the user to respond
with various answers. There is a DEMONSTRATION option
available that will provide first time users with an
overview of the functions of the simulator.  More experienced
users can exercise more control over the operation of the
simulator through the USER SPECIFIED option where specific
simulation controls and grinding circuit constants can be
interactively chosen. The required response format is indicated
in the example that accompanies the input prompt.  When
choosing an item from a menu enter an INTEGER number and
press return/enter.  Other numeric input is required in a
REAL variable format or unpredictable results may occur.
Yes/No responses default an appropriate value depending on
the specific situation.

         Continue? (Y/N)

Y
    The user should be aware of the following items
with respect to the cost of the simulation:
1. Simulation time is the most critical cost variable;
   the longer the simulation the higher the cost!
2. Integration interval is the next most critical cost factor
   and interacts with the simulation time; in general, the
   smaller the integration interval, the higher the cost!
3. The print interval also plays a key role in determining
   simulation cost because data printing is expensive; the
   smaller the print interval, the higher the cost!
4. In all cases the LINEAR MODEL is less costly and the DEMO
   simulator always uses this model.
         Continue? (Y/N)
```

```
Y


                          SIMULATION TYPE
                          ---------------

                   -----------------------------
                        1 - Demonstration
                        2 - User Specified
                   -----------------------------

    CHOOSE A SIMULATION TYPE BY NUMBER
2

                          SIMULATION CIRCUIT
                          ------------------

             --------------------------------------------
                  1 - Lake Dufault Circuit: Open Loop
                  2 - Lake Dufault Circuit: Type I Control
                  3 - Lake Dufault Circuit: Type II Control
                  4 - Lake Dufault Circuit: TI (Multi - loop)
                  5 - Lake Dufault Circuit: TII (Multi - loop)
                  6 - Lake Dufault Circuit: TI with TDC
                  7 - Lake Dufault Circuit: TII with TDC
                  8 - Lake Dufault Circuit: TI with STC
                  9 - Lake Dufault Circuit: TII with STC
             --------------------------------------------

    CHOOSE A SIMULATION CIRCUIT BY NUMBER
2


                            MODEL TYPE
                            ----------

                      ----------------------
                       1 - Nonlinear Model
                       2 - Linear Model
                      ----------------------

         CHOOSE A MODEL TYPE BY NUMBER
1


    ENTER INTEGRATION INTERVAL?
.1

    ENTER SIMULATION TIME?
20.

    ENTER PRINT INTERVAL?
1.


    ENTER LOOP 1 CONTROLLER CONSTANTS
    (GAIN, INTEGRAL TIME, SETPOINT - e.g. 10.,2.0,81.048)
10.,2.,81.048

    ENTER LOOP 2 CONTROLLER CONSTANTS
    (GAIN, INTEGRAL TIME, SETPOINT - e.g. 82.3,1.385,.95573)
82.3,1.385,.95573
```

```
                         CLOSED LOOP DISTURBANCE TYPE
                        -------------------------------

                         -----------------------------
                           1 - Loop 1 Setpoint Step
                           2 - Loop 2 Setpoint Step
                           3 - Ore Hardness Step
                         -----------------------------

    CHOOSE A DISTURBANCE BY NUMBER
3

    ENTER MAGNITUDE OF DISTURBANCE IN %
50.

    ENTER TIME OF DISTURBANCE?
2.


                    SUMMARY OF SIMULATION PARAMETERS

                 DYNAMIC SIMULATION OF A GRINDING CIRCUIT
                 -----------------------------------------
                            TYPE I CONTROL
                         Lake Dufault Circuit
     -------------------------------------------------------------
     -------------------------------------------------------------
                            Nonlinear Model


    TOTAL SIMULATION TIME =  20.00 MINUTES
    INTEGRATION INTERVAL =   0.10 MINUTES

    LOOP 1 CONTROLLER CONSTANTS
      GAIN =      10.000
      INTEGRAL TIME =       2.000
      SETPOINT = 81.0480
      SETPOINT STEP CHANGE =  0.0    AT TIME   0.0

    LOOP 2 CONTROLLER CONSTANTS
      GAIN =      82.300
      INTEGRAL TIME =       1.385
      SETPOINT =  0.9557
      SETPOINT STEP CHANGE =  0.0    AT TIME   0.0

    ORE HARDNESS DISTURBANCE INFORMATION
      DISTURBANCE MAGNITUDE =    50.0000
        AT TIME =   2.000 (ROD MILL)
           TIME =   8.000 (BALL MILL)


     IS THE ABOVE INFORMATION CORRECT? (Y/N)
Y



          WORKING ...



          ... SIMULATION COMPLETED



 Results are in temporary file "-MIN.OUT"
#$COPY -MIN.OUT
```

G.2 Lake Dufault Grinding Circuit Simulator Sample Output

```
            DYNAMIC SIMULATION OF A GRINDING CIRCUIT
            ----------------------------------------
                         TYPE I CONTROL
                      Lake Dufault Circuit
 -----------------------------------------------------------
 -----------------------------------------------------------
                         Nonlinear Model

 TOTAL SIMULATION TIME =  20.00 MINUTES
 INTEGRATION INTERVAL =   0.10 MINUTES

 LOOP 1 CONTROLLER CONSTANTS
   GAIN =      10.000
   INTEGRAL TIME =       2.000
   SETPOINT = 61.0480
   SETPOINT STEP CHANGE =  0.0    AT TIME   0.0

 LOOP 2 CONTROLLER CONSTANTS
   GAIN =      52.300
   INTEGRAL TIME =       1.386
   SETPOINT =  0.5557
   SETPOINT STEP CHANGE =  0.0    AT TIME   0.0

 ORE HARDNESS DISTURBANCE INFORMATION
   DISTURBANCE MAGNITUDE =      50.0000
     AT TIME =   2.000 ([illegible] MILL)
        TIME =   6.000 (BALL MILL)



ROD MILL SPECIFICATIONS FOLLOW...
---------------------------------
PULP HOLDUP VOLUME(M**3)= 3.8

LOWER             FIRST ORDER    FIRST COLUMN
BOUND             SELECTION      OF BREAKAGE
ON SIZE           CONSTANT       MATRIX
---------------------------------------------
    4757.           22.950           .0
    3364.            4.660          .2080
    2378.            1.028          .1850
    1682.            0.218          .1310
    1189.            0.048          .1030
     841.            0.021          .0820
     595.            0.021          .0650
     420.            0.021          .0510
     297.            0.021          .0410
     210.            0.021          .0320
     149.            0.021          .0250
     105.            0.021          .0200
      74.            0.021          .0160
      53.            0.021          .0130
      37.            0.021          .0100
      10.            0.0            .0
---------------------------------------------

BALL MILL SPECIFICATIONS FOLLOW...
----------------------------------
PULP HOLDUP VOLUME(M**3)=15.8

LOWER             FIRST ORDER    FIRST COLUMN
BOUND             SELECTION      OF BREAKAGE
ON SIZE           CONSTANT       MATRIX
---------------------------------------------
    4757.           17.804           .0
    3364.            8.463          .2020
    2378.            5.047          .1810
    1682.            2.880          .1290
    1189.            1.432          .1030
     841.            0.763          .0820
     595.            0.407          .0650
     420.            0.216          .0520
     297.            0.115          .0420
     210.            0.061          .0330
     149.            0.033          .0270
     105.            0.017          .0210
      74.            0.009          .0170
      53.            0.005          .0130
      37.            0.003          .0110
      10.            0.0             0
---------------------------------------------
```

```
STREAM ATTRIBUTES AT TIME   0.0
.................................
SIZE       STRM  1 STRM  2 STRM  3 STRM  4 STRM  5 STRM  6 STRM  7 STRM  8 STRM  9 STRM 10 STRM 11 STRM 12 STRM 13 STRM 14
.............................................................................................................................
  4757.    61.00    61.00     0.0    61.00     0.44     0.44     0.0      0.15     0.0      0.15     0.0      0.24     0.00     0.00
  3364.     8.00     8.00     0.0     8.00     0.88     0.88     0.0      0.24     0.0      0.24     0.0      0.37     0.00     0.00
  2378.     6.00     6.00     0.0     6.00     2.80     2.80     0.0      1.00     0.0      1.00     0.00     1.54     0.03     0.03
  1682.     5.24     5.24     0.0     5.24     8.88     8.88     0.0      3.15     0.0      3.15     0.00     4.85     0.18     0.18
  1189.     2.84     2.84     0.0     2.84    13.27    13.27     0.0      4.98     0.0      4.98     0.01     7.68     0.51     0.51
   841.     1.80     1.80     0.0     1.80    12.19    12.19     0.0      4.98     0.0      4.98     0.11     7.57     1.05     1.05
   595.     1.50     1.50     0.0     1.50    10.10    10.10     0.0      4.83     0.0      4.83     0.42     7.20     1.88     1.88
   420.     1.04     1.04     0.0     1.04     8.01     8.01     0.0      5.11     0.0      5.11     1.16     7.22     3.54     3.54
   297.     1.08     1.08     0.0     1.08     6.80     6.80     0.0      6.31     0.0      6.31     2.85     8.18     5.05     5.05
   210.     1.18     1.18     0.0     1.18     5.79     5.79     0.0      7.94     0.0      7.94     5.82     9.18     9.08     9.08
   149.     1.21     1.21     0.0     1.21     4.98     4.98     0.0      9.14     0.0      9.14     8.75     9.34    11.38    11.38
   105.     1.17     1.17     0.0     1.17     4.23     4.23     0.0      9.20     0.0      9.20    10.88     8.30    11.87    11.87
    74.     1.13     1.13     0.0     1.13     3.64     3.64     0.0      8.27     0.0      8.27    11.24     6.88    10.77    10.77
    53.     1.38     1.38     0.0     1.38     3.40     3.40     0.0      7.12     0.0      7.12    10.82     5.23     9.13     9.13
    37.     0.52     0.52     0.0     0.52     2.27     2.27     0.0      5.40     0.0      5.40     8.82     3.68     7.08     7.08
    10.     4.84     4.84     0.0     4.84    12.75    12.75     0.0     22.23     0.0     22.23    39.71    12.80    27.34    27.34
.............................................................................................................................
MF SOLIDS  65.40    65.40     0.0    65.40    65.40    65.40     0.0    186.67     0.0    186.67    65.40   121.27   121.27   121.27
MF WATER    0.0      0.0     11.54    11.54    11.54    11.54    47.78    91.04     0.0     91.04    59.33    31.71    31.71    31.71
VOL. FLOW  27.25    27.25    11.54    38.78    38.78    38.78    47.78   168.82     0.0    168.82    86.58    82.24    82.24    82.24
% SOLIDS  100.00   100.00     0.0    85.00    85.00    85.00     0.0     67.22     0.0     67.22    52.43    79.27    79.27    79.27
.............................................................................................................................




SIMULATION                   CYCLONE      PUMP         SUMP         BALL MILL    BALL MILL    FRESH ORE    CYCLONE      CYCLONE
TIME IN                      FEED         SUMP         FEED WATER   DISCHARGE    DISCHARGE    FEED RATE    OVERFLOW     OVERFLOW
MINUTES                      DENSITY      HEIGHT       FLOWRATE     -150 MIC     SOLIDS MASS               -150 MIC     SOLIDS MASS
                             (%)          (M)                       FREQUENCY    FLOWRATE                  FREQUENCY    FLOWRATE
...................................................................................................................................
 0.0           0.0           6.7218E+01   5.5574E-01   4.7780E+01   2.7335E+01   1.2127E+02   6.5400E+01   8.1048E+01   6.5400E+01
 1.0000E+00    0.0           6.7218E+01   5.5573E-01   4.7791E+01   2.7335E+01   1.2127E+02   6.5404E+01   8.1048E+01   6.5400E+01
 2.0000E+00    0.0           6.7218E+01   5.5573E-01   4.7791E+01   2.7335E+01   1.2127E+02   6.5404E+01   8.1048E+01   6.5400E+01
 3.0000E+00    0.0           6.7218E+01   5.5573E-01   4.7792E+01   2.7335E+01   1.2127E+02   6.5404E+01   8.1048E+01   6.5400E+01
 4.0000E+00    0.0           6.7223E+01   5.5574E-01   4.7722E+01   2.7334E+01   1.2137E+02   6.5470E+01   8.1054E+01   6.5324E+01
 5.0000E+00    0.0           6.7236E+01   5.5574E-01   4.7629E+01   2.7332E+01   1.2155E+02   6.5475E+01   8.1050E+01   6.5206E+01
 6.0000E+00    0.0           6.7258E+01   5.5574E-01   4.7607E+01   2.7328E+01   1.2174E+02   6.5374E+01   8.1041E+01   6.5085E+01
 7.0000E+00    0.0           6.7276E+01   5.5573E-01   4.7633E+01   2.7322E+01   1.2191E+02   6.5220E+01   8.1031E+01   6.4993E+01
 8.0000E+00    0.0           6.7267E+01   5.5573E-01   4.7660E+01   2.7318E+01   1.2205E+02   6.5057E+01   8.1025E+01   6.4898E+01
 9.0000E+00    0.0           6.7293E+01   5.5572E-01   4.7715E+01   2.6793E+01   1.2216E+02   6.4911E+01   8.1023E+01   6.4812E+01
 9.9999E+00    0.0           6.7268E+01   5.5570E-01   4.7694E+01   2.6290E+01   1.2236E+02   6.4425E+01   8.0985E+01   6.4579E+01
 1.1000E+01    0.0           6.7232E+01   5.5568E-01   4.8408E+01   2.5864E+01   1.2266E+02   6.3371E+01   8.0922E+01   6.4059E+01
 1.2000E+01    0.0           6.7120E+01   5.5564E-01   4.9057E+01   2.5472E+01   1.2291E+02   6.2188E+01   8.0876E+01   6.3388E+01
 1.3000E+01    0.0           6.6984E+01   5.5562E-01   4.9906E+01   2.5113E+01   1.2309E+02   6.1066E+01   8.0858E+01   6.2800E+01
 1.4000E+01    0.0           6.6793E+01   5.5562E-01   5.0172E+01   2.4782E+01   1.2321E+02   6.0045E+01   8.0854E+01   6.1798E+01
 1.5000E+01    0.0           6.6603E+01   5.5562E-01   5.0894E+01   2.4473E+01   1.2328E+02   5.9119E+01   8.0858E+01   6.0985E+01
 1.6000E+01    0.0           6.6438E+01   5.5563E-01   5.1179E+01   2.4164E+01   1.2332E+02   5.8286E+01   8.0868E+01   6.0184E+01
 1.7000E+01    0.0           6.6218E+01   5.5564E-01   5.1808E+01   2.3912E+01   1.2332E+02   5.7578E+01   8.0885E+01   5.9424E+01
 1.8000E+01    0.0           6.6028E+01   5.5564E-01   5.2008E+01   2.3656E+01   1.2332E+02   5.6905E+01   8.0887E+01   5.8702E+01
 1.9000E+01    0.0           6.5848E+01   5.5565E-01   5.2391E+01   2.3414E+01   1.2331E+02   5.6270E+01   8.0907E+01   5.8011E+01
 2.0000E+01    0.0           6.5672E+01   5.5565E-01   5.2742E+01   2.3168E+01   1.2328E+02   5.5712E+01   8.0920E+01   5.7384E+01




STREAM ATTRIBUTES AT TIME  20.00
.................................
SIZE       STRM  1 STRM  2 STRM  3 STRM  4 STRM  5 STRM  6 STRM  7 STRM  8 STRM  9 STRM 10 STRM 11 STRM 12 STRM 13 STRM 14
.............................................................................................................................
  4757.    61.00    61.00     0.0    61.00     0.74     0.75     0.0      0.24     0.0      0.28     0.0      0.38     0.00     0.00
  3364.     8.00     8.00     0.0     8.00     1.14     1.15     0.0      0.37     0.0      0.38     0.0      0.56     0.01     0.01
  2378.     6.00     6.00     0.0     6.00     4.38     4.42     0.0      1.43     0.0      1.48     0.00     2.17     0.08     0.08
  1682.     5.24     5.24     0.0     5.24    11.28    11.32     0.0      3.78     0.0      3.87     0.00     5.67     0.38     0.39
  1189.     2.84     2.84     0.0     2.84    13.74    13.73     0.0      4.97     0.0      5.02     0.01     7.38     1.00     1.01
   841.     1.80     1.80     0.0     1.80    11.78    11.74     0.0      4.99     0.0      5.01     0.08     7.31     1.84     1.85
   595.     1.50     1.50     0.0     1.50     9.84     9.83     0.0      5.32     0.0      5.28     0.33     7.55     3.41     3.36
   420.     1.04     1.04     0.0     1.04     7.54     7.52     0.0      5.93     0.0      5.77     1.10     7.84     5.30     5.22
   297.     1.08     1.08     0.0     1.08     6.38     6.34     0.0      7.18     0.0      7.00     2.88     8.92     7.70     7.59
   210.     1.18     1.18     0.0     1.18     5.38     5.37     0.0      8.58     0.0      8.40     5.78     9.63    10.11    10.02
   149.     1.21     1.21     0.0     1.21     4.58     4.58     0.0      9.33     0.0      9.24     8.98     9.37    11.52    11.48
   105.     1.17     1.17     0.0     1.17     3.90     3.88     0.0      8.98     0.0      8.87    11.04     9.01    11.27    11.28
    74.     1.13     1.13     0.0     1.13     3.34     3.33     0.0      7.82     0.0      7.88    11.38     6.23     9.80     9.84
    53.     1.38     1.38     0.0     1.38     3.16     3.14     0.0      6.60     0.0      6.68    10.70     4.78     8.11     8.18
    37.     0.52     0.52     0.0     0.52     2.01     2.00     0.0      4.80     0.0      4.88     8.80     3.28     6.15     6.20
    10.     4.84     4.84     0.0     4.84    11.09    11.08     0.0     19.59     0.0     19.89    38.22    10.89    23.18    23.40
.............................................................................................................................
MF SOLIDS  55.71    55.71     0.0    55.71    55.71    55.71     0.0    179.13     0.0    180.65    57.38   123.28   123.28   123.42
MF WATER    0.0      0.0      9.83     9.84     9.84     9.84    52.74    95.06     0.0     94.43    61.88    32.55    32.55    32.49
VOL. FLOW  23.21    23.21     9.83    33.05    33.05    33.05    52.74   169.70     0.0    169.70    85.78    83.92    83.92    83.92
% SOLIDS  100.00   100.00     0.0    84.99    84.99    84.99     0.0     65.33     0.0     65.67    48.11    79.11    79.11    79.16
.............................................................................................................................
```

G.3 Open Circuit Grinding Simulator Sample Session

```
#SO GMCD:MINSIM.3.EXE
#SET MACROS=ON
#MINSIM3
#SET ECHO=OFF


                      OPEN-CIRCUIT GRINDING SIMULATOR
                       After Adel, Ulsoy and Sastry,
        A Theoretical Analysis and Control Study of Open Circuit Grinding
                       Int. J. Min. Proc. 10, 1, 1983
                                COPYRIGHT
                          University of Alberta
                     Department of Chemical Engineering
                        Written by: G.B. McDougall
 ---------------------------------------------------------------------

                          INTERACTIVE VERSION 3.1


   Continue? (Y/N)
Y


   ENTER INTEGRATION INTERVAL?
.1

   ENTER TOTAL SIMULATION TIME?
10.

   ENTER PRINT INTERVAL?
.5


                             CONTROL SYSTEM
                            ----------------
                             1 - Open Loop
                             2 - Continuous
                             3 - Discrete
                            ----------------

   Choose a Control System By Number

2


                             CONTROLLER MODE
                            ------------------
                               1 - P only
                               2 - P+I
                               3 - P+I+D
                            ------------------

   Choose a Controller by Number
```

```
2
    ENTER PROPORTIONAL GAIN

90.

    ENTER INTEGRAL GAIN

27.



                          Disturbance
               ----------------------------------------
                1 - Grindability/Surface Area Step
                2 - Grindability/Surface Area Pulse Train
               ----------------------------------------

    Choose a System Disturbance By Number

1
    Time for Disturbance to Enter System?

0.

    Enter Analysis Delay?

1.


        ...Done

        Output in -MIN.OUT

#SC -MIN.OUT
```

G.4 Open Circuit Grinding Simulator Sample Output

```
                    OPEN-CIRCUIT GRINDING SIMULATOR
                     After Adel, Ulsoy and Sastry,
A Theoretical Analysis and Control Study of Open Circuit Grinding
                    Int. J. Min. Proc. 10, 1, 1983
                              COPYRIGHT
                         University of Alberta
                 Department of Chemical Engineering
                      Written by: G.B. McDougall


                       INTERACTIVE VERSION 3.1


-------------------------------------------------------------


Total Simulation Time =    10.00 hours
Integration Interval =    0.100 hours


Continuous Control
Proportional Integral Mode
Proportional Gain =  90.000
Integral Gain =   27.000


Disturbance Information
Feed Grindability/Surface Area Step Change
Entering System at Time =     0.0  hours

Analysis Delay =    1.000 hours
```

```
TIME                   FEED         FEED       FEED       OUTPUT        TIME
                   GRINDABILITY  SURFACE       RATE      SURFACE     AVERAGED
                                   AREA                    AREA        ERROR
(h)                (m**2/kWh)   (m**2/kg)     (kg/h)    (m**2/kg)    (m**2/kg)

-------------------------------------------------------------------------------------

0.0         0.0     3.0000E+03  2.2500E+01  0.0         0.0         0.0         0.0
5.0000E-01  0.0     3.0000E+03  2.2500E+01  0.0         9.4584E+00  6.0179E+00  0.0
1.0000E+00  0.0     3.0000E+03  2.2500E+01  0.0         1.5446E+01  9.6442E+00  0.0
1.5000E+00  0.0     3.0000E+03  2.2500E+01  1.0365E+03  1.6407E+01  1.2014E+01  0.0
2.0000E+00  0.0     3.0000E+03  2.2500E+01  1.8387E+03  1.3517E+01  1.2790E+01  0.0
2.5000E+00  0.0     3.0000E+03  2.2500E+01  2.1885E+03  9.6872E+00  1.2428E+01  0.0
3.0000E+00  0.0     3.0000E+03  2.2500E+01  2.1296E+03  7.1947E+00  1.1625E+01  0.0
3.5000E+00  0.0     3.0000E+03  2.2500E+01  1.9199E+03  6.4802E+00  1.0863E+01  0.0
4.0000E+00  0.0     3.0000E+03  2.2500E+01  1.7953E+03  6.7558E+00  1.0305E+01  0.0
4.5000E+00  0.0     3.0000E+03  2.2500E+01  1.8265E+03  7.0274E+00  9.9174E+00  0.0
5.0000E+00  0.0     3.0000E+03  2.2500E+01  1.9560E+03  6.7786E+00  9.6049E+00  0.0
5.5000E+00  0.0     3.0000E+03  2.2500E+01  2.0903E+03  6.0665E+00  9.2964E+00  0.0
6.0000E+00  0.0     3.0000E+03  2.2500E+01  2.1723E+03  5.2269E+00  8.9679E+00  0.0
6.5000E+00  0.0     3.0000E+03  2.2500E+01  2.1996E+03  4.5466E+00  8.6299E+00  0.0
7.0000E+00  0.0     3.0000E+03  2.2500E+01  2.2016E+03  4.1109E+00  8.3025E+00  0.0
7.5000E+00  0.0     3.0000E+03  2.2500E+01  2.2078E+03  3.8417E+00  7.9979E+00  0.0
8.0000E+00  0.0     3.0000E+03  2.2500E+01  2.2301E+03  3.6159E+00  7.7175E+00  0.0
8.5000E+00  0.0     3.0000E+03  2.2500E+01  2.2641E+03  3.3576E+00  7.4560E+00  0.0
9.0000E+00  0.0     3.0000E+03  2.2500E+01  2.2987E+03  3.0610E+00  7.2078E+00  0.0
9.5000E+00  0.0     3.0000E+03  2.2500E+01  2.3264E+03  2.7610E+00  6.9696E+00  0.0
9.9999E+00  0.0     3.0000E+03  2.2500E+01  2.3459E+03  2.4926E+00  6.7412E+00  0.0
```